School of Business,
Economics and Law
GÖTEBORG UNIVERSITY
The Department of Informatics

Publish date 2006-06-20

SERVICE ORIENTED ARCHITECTURE;
THE IMPORTANCE OF SERVICE REGISTRY
A CASE STUDY AT IBM

ABSTRACT

The challenge to manage the alignment of business and IT development have struggled with poor results, it is essential to bridging the gap between line of business and IT. Spaghetti structure are a well known phenomena in information systems development, this appears when several systems are integrated with point-to-point solutions. The architectural development from client-server architecture and network centric architecture are now moving toward service oriented architecture (SOA). Lack of control could eventually lead to another level of spaghetti structure. Especially in a global world characterized by inter-organizational and interconnected IT and business environments driven by unpredictable customer behavior. Today the aim for management is not only to improve efficiency but also to improve business effectiveness and to manage organizations and customers behaviors more strategically. Our objective is to devise and illustrate the importance of controlling, survey and manage services in service oriented architectures. Furthermore, to increase the knowledge of people engaged in IT and business development regarding service oriented architecture and the concept service registry. The scientific approach was semi structured interviews with a number of respondents, for the thesis, relevant positions in the organization. The implication provides identified key success factors for comprehension of services in SOA and managing balance of dependencies in changing information environments characterized by SOA. We claim that we have created a foundation for an expanded meaningfulness of SOA service, which will lead to higher awareness on improved overview, governance and mutual understanding of SOA services.

KEYWORDS

SOA, Service Registry, IT management, IT governance, SOA governance, Separation of Concerns, Life Cycle Management, Spaghetti structure, Service

This thesis is written in English

THERESE OLOFSSON AND JOHANNES SVENSSON
Academic supervisor: Maria Bergenstjerna
Master thesis 20 p.

ACKNOWLEDGEMENT

We would like to show gratitude to IBM and especially thank all respondents that gave us interesting insights on the issue. Regarding tips and thoughts we would like to thank Professor Bo Dahlbom. Also, we would like to direct an enormous thank you to our academic supervisor Maria Bergenstjerna for giving us constructive criticism and for her dedication to our thesis; we could not have done this without you. We specially would like to show thankfulness towards our two contacts at IBM Software Group: Berny Hammarlund and Nils Bergquist; we are truly grateful for your help and commitment to our work, also your precious thoughts and ideas. Last but certainly not least, we would like to thank our friends and families for their opinions and their support.

Thank you all once again for your dedication!

Gothenburg, June 2006

Therese Olofsson Johannes Svensson

TABLE OF CONTENT

TABLE OF FIGURES

1 INTRODUCTION

This Master thesis is the result of the final twenty weeks of the System Engineering and IT management program at School of Business, Economics and Law, Gothenburg University. The mission is to give us an opportunity to use our knowledge, find new relevant information, and work in a scientific way. Our interest for the study topic started with a guest lecture by Professor Bo Dahlbom, he introduced us to service oriented architecture (SOA). Our educational background, which will lead to a Masters in Informatics, made us more interested in this subject area and we decided to contact IBM, who in some way initiated the concept; service oriented architecture. In numerous discussions with our contacts within IBM more questions came up, especially that lack of service control could eventually lead to another level of spaghetti structure, we call it service spaghetti. Spaghetti structures are a well know phenomena in information systems development. Spaghetti structures appear when several systems are connected together as point-to-point solutions. The connections between systems increase exponentially when new systems are connected and we have no longer overview of the connections. In the following chapter we will in detail explain what we mean by service spaghetti.

1.1 Background

The aim for management today is not only to improve efficiency but also to improve business effectiveness and manage organizations and costumers behavior more strategically. As decision-making tasks become more complex, so does the required information - from structured, on a daily basis support to, unstructured, complex enquiries at the highest levels of management. (Galliers & Leidner, 2003)

Elderly systems have a tendency to bring problems when organizations make upgrades in business activities and in the IT environment (Galliers & Leidner, 2003). Magoulas and Pessi (1998) more specified describe that the problem is based on the world of information systems that exists in several larger organizations characterized by problematic information labyrinths, information islands and information bureaucracy. These three expressions are according to Magoulas and Pessi, (1998) commonly used to describe bad information system architectures and organizations today become more and more dependent on information systems (IS) and information technology (IT) to reach success.

Further state Magoulas and Pessi (1998) that within the business world there are high demands on fast and effective adjustment to a dynamic and complex business world. Organizational information systems estimate on these occasions to play a big part managing those requirements. Organizations have integrated existing systems with new systems for a long time, if this increase with more and more systems it can without doubt lead to so called information labyrinths. Spaghetti structure and information islands are two other existing terms. These terms are defined by several information systems integrated in a non overviewed way and very difficult to control. They can also be defined by separated information systems characteristic by duplicated information in an anarchistic way. (Hewitt et al., 1982; Magoulas & Pessi, 1998)

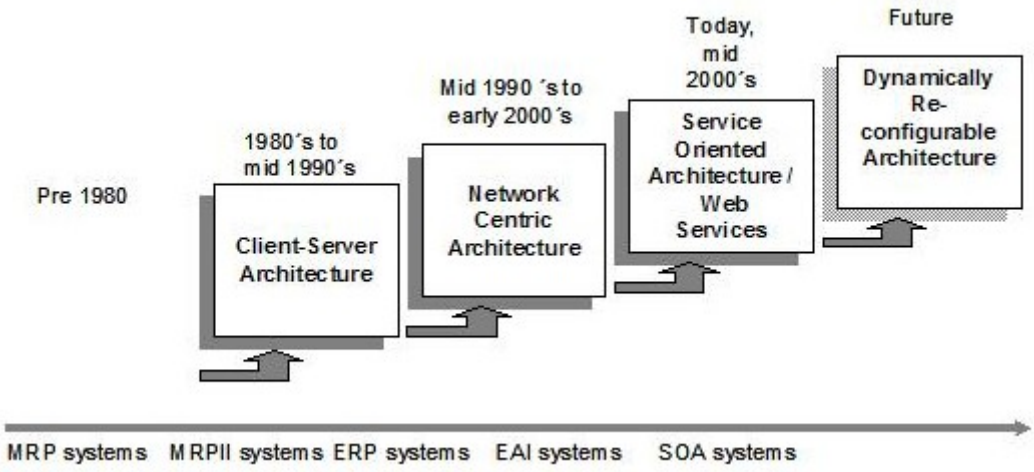


Figure 1. Information Systems Architectural evolution in the enterprise, figure inspired by Magnusson and Olsson (2005) and IBM internal presentation (2006)

Magoulas and Pessi (1998) declare that the concept of information and information system seems to be the most widely spread notion in our information community. Many organizations have invested a large amount of money on the development of information systems and many others will follow. The modern information community requires information systems, because they enable supervision and distribution of information. Creating information systems is not a problem free activity, so, for that reason new academic disciplines will appear and guide us through systematization and control of information, and there by control of the information systems.

A well designed structure of the enterprise is not a guarantee to solve a specific problem in an activity. Thus, it helps to create better conditions to effective handle those problems. Difficulties that occurs with business and systems integration has been a known problem for decades, several articles and books support this statement (Earl, 1989; Hammer & Champy, 1993; Höij, 2006; Khosrowpour, 1997; Magnusson & Olsson, 2005; Magoulas & Pessi, 1998) likewise in conversation with companies, not only in the IT-business but with any organization which has integrated or intend to integrate enterprise solutions. To solve the problem with information and architectural spaghetti and islands of information many vendors of IT solutions state that the answer to this could be Service Oriented Architecture (SOA). The heart of SOA is services, independent components of information that realize services and service flows. Owens (2006) say that in the past, when solution architects asked "How can we best integrate multiple, disparate applications and use this to develop new services?" the answer was Web services. The new answer is progressively becoming SOA. However, Owens (2006) state that using Web services does not necessarily mean that there is a SOA, and having a SOA does not automatically state the use of Web services. But one thing a SOA might require is a registry for service storage.

Höij (2006) state that it clearly exist a solutions to the problem with spaghetti structures and information islands, but another problem can occur when using SOA to integrate applications, according to the history of information islands you can say that the spaghetti structure can become service spaghetti on a higher abstract conceptual level. Owens (2006) state that service registry is necessary in a SOA to avoid this problem. Ort (2005) says that manage services is not only a solution; it is also a cornerstone of SOA governance. It gives

organizations a registry for publishing and discovering needed information about services. With that organizations will be provided a centralized way of seeking and managing information about services and also related substance based on things like life-cycle stage and organization policies. Still, with this, state Fox (2006) that SOA will fail if the organizations operators do not sit down and inspect the enterprise processes with IT architects and investigate what information is essential.

1.2 Purpose and Issue

The main purpose with this thesis is to devise and illustrate the importance of controlling, survey and manage services in service oriented architectures. Furthermore, to increase the knowledge regarding the concept of service oriented architecture and strengthen the knowledge of the concept service registry by study it in the actual context as well as discuss and analyze the concepts existence and argue around different consequences the idea has on the subject of IT management. Moreover, to identify key success factors for comprehension of services in SOA. With this, a foundation lay for an expanded meaningfulness of SOA service, which will lead to higher awareness and improved overview, governance and mutual understanding of SOA services to an increased clientele.

1.2.1 Hypothesis

The old way of thinking has led to complicated IT environments and postponement of enterprise development rather than support it. The architectural development from client-server architecture and network centric architecture are now moving toward service oriented architecture. When using SOA without considerations about earlier problems with architectural spaghetti a similar problem might lead to another level of architectural spaghetti, like we stated before, we choose to call it; *service spaghetti.* On the basis of these problems, information's and thoughts we designed this question at issue and attempt to investigate;

Which key success factors can be identified for managing balance of dependencies in changing information environments characterized by SOA?

1.3 Delimitation

When writing a thesis, it is of importance to limit the study. This is suitable to the fact that the researcher needs to focus on particular issues within the research problem area, instead of describing all possible problem areas. If we would not limit, the thesis would probably be unfocused, lack relevance and also become too widespread. In this study, neither questions of deeper technical character, specific cost benefits analysis nor methods on how to measure these are explicitly analyzed. Technical concepts are treated as an important factor but we do not in detail declare specific technical equipments.

We understand that cost is an essential factor that has to be considered in every decision process. However we find it difficult to furthermore analyze costs because of the time limit, for example issues as; when not using service registry in SOA. We are aware of the fact that this might limit the usefulness of our study, since many IT management decisions are dependent on costs and technical equipment. Also we consider IBM as a company like every other company and refrain from involving IBM's sales division. Within the concept IT Governance there are organizations or institutes which focus on standards in directing and controlling an enterprise's information technology. They have developed frameworks for effective IT governance such as CobiT, ITIL and ISO 17799. This thesis has no intention of

describing them or deeper explain them. We just mention them here to inform the reader about their existence.

1.3.1 Expected readers

Expected readers of this Master thesis are people working with information technology and IT management, employees of IBM. Also, informatics students as well as other people with interest in service registry, SOA and IT management. This thesis encloses a definition glossary in section 2.1 were we define some of the main concepts. Our academic ambitions are high and this is the proof and conclusion of our four years of studies within the area of Informatics.

1.4 Disposition

Chapter [1] – *Introduction*
This chapter includes the background in addition to the problem area. We present the purpose and issue also the question at issue; the hypothesis. The chapter also includes delimitation.

Chapter [2] –*Theoretical frame of reference*
This chapter contains structured perusal of current literature on the research topic, which leads to the scientific framework for the research. The purpose is to provide a deeper understanding of the concept and the problem area. The chapter also present our theoretical model based on fundamental parts concerning this thesis such as IT management, IT governance, SOA Governance, Separation of concerns, Service oriented architecture (SOA), Service management, Service life cycle, Quality and quality of Service and Registry.

Chapter [0] – *Scientific approach*
In this section we briefly summarize the proceedings of our research process and present an introduction to management research and an explanation of what qualitative and quantitative methods are. Based on our labor and accomplishment we have designed a model (see Figure 9) for the master thesis course of action; were the pieces of the thesis are connected and related to each other in one way or another, the model is presented in this chapter.

Chapter [4] – *Case study result*
This chapter presents the case study based on interviews with respondents at IBM.

Chapter [5] – *Analysis and Discussion*
In this chapter we analyze and discuss our theoretical frame of reference and our case study. We present the analysis by categorize the respondents professional position and the questions in tree respectively five categories. This is done because we aim to widen and increase the knowledge of SOA on different business levels, also to see if the opinions differ from each level.

Chapter [6] – *Conclusion and Recommendations*
Conclusions are drawn and presented in this chapter and principally the hypothesis for this thesis is answered. This section also contains recommendations to expected readers and of course IBM.

2 THEORETICAL FRAME OF REFERENCE

This chapter contains structured perusal of current literature on the research topic, which leads to the scientific framework for the research. The purpose is to provide a deeper understanding of the concept and the problem area. To facilitate the reading and the comprehension of the theory we have enclosed various sections in this chapter with a chapter summary and highlights.

2.1 Definition of basic concepts

Some of the concepts that are used in this thesis are not easy to define, and often applied in different ways depending on situation. We present some of the definitions once again later in each chapter just to make clear that we intend to increase knowledge and amplify the comprehension.

- *Organization*: Organization derives from the Greek *organon*, meaning an intellectual tool or an instrument to arrange something (knowledge or human activities or ideas etc.). An organization is traditionally seen as one single company that operates in one market. Today the truth is somewhat different; more then one company can form an organization as a result of a merger or when two companies form a partnership to e.g. gain competitive edge. Organizations are formed entities that work towards a common goal through agreement and they are becoming more independent of geographical borders. The organization has no subsistence without their members (Ansoff, Declerck & Hayers, 1976; Earl, 1989).
- *Information systems (IS)* mean an interconnected set of information resources under the same direct management control that shares common functionality. A system normally includes hardware, software, information, data, applications, communications, and people (ICHNET, 2005; Johnston & Bowen, 2005)
- *Separation of concerns* (SoC): Is a general principle in software engineering introduced by Dijkstra, (1976) and Parnas (1972) as an answer to control the complexity of ever growing programs. The purpose and benefits of such modular approaches include reduction of complexity, ease of updates and changes, reuse, simplifying customization.
- *Product life cycle management* (PLC Management) - Most discussions of product life cycle (PLC) portray the history of typical product as following an S-shaped curve. The curve is typically divided into four stages, known as *introduction*, *growth*, *maturing* and *decline*. Each performance form, organization, or specific offering, and all levels of the product may be subjected to the cycle analysis (Kotler et al., 1997).
- *IT management* (IM) strategy: This concept comprises the policies, procedures, aims and actions likely to be identified from using IS/IT, its role and the structure of IT activities within the organization. IT management focuses on the relationship between management and end users (Magoulas & Pessi, 1998).
- *IT governance*: IT Governance can be defined as the schema for specifying decision rights, accountability and responsibility used by organizations to achieve a desirable behavior in the use of information technology (IT). Enterprises should not use IT governance to make specific IT decisions, the management does that. Instead IT governance determines systematically who should make and who should contribute in management decisions (Ross & Weill, 2004).

- *SOA governance*: Design rules combined with enforcement are typically called *policies*. The development and enforcement of SOA policies and procedures goes by the name SOA governance. Governance and architecture go hand in hand. SOA governance provides context for system architects and designers (Java Sun, 2005) and SOA governance is an extension of IT governance and is used in the context of service oriented architecture. The objective is to create a sufficient level of desirable behaviors. SOA governance is social in nature, inviting continuous dialog between developers and architects (Weill & Ross, 2004).
- *SOA*- In service oriented architecture, a system operates as a collection of services. Each service may interact with various other services to accomplish a certain task. The operation of one service might be a combination of several low level functions, e.g. functions that converts objects to basic data types, and in this case, these level functions are not considered as services (Erl, 2005).
- *Service* - Services are all of the organizational activities and processes an enterprise can identify and further divide into sub processes such as business activities, business transactions and system activities. Services work together in a network of software components that can receive a call and provide a given behavior on behalf of the calling system, application or component. One should especially note the difference between services. *Web services* are a set of technologies and standards for interoperability well adapted to exploit the characteristics and potentials of the Web environment. The notion of service is of special importance in this context: it refers to a network addressable software component (Cordeiro & de Carvalho, 2002).
- *Quality of Service* - Five distinct, literal, holographic dimensions of quality provide an innovative framework for assessing, planning, and implementing service quality. Service quality, at its very best, is fulfilled through quality's five dimensions: Experience brings vision into reality. Measurement provides knowledge of the system. Relationships and Systems Thinking help an organization become "free of mortal risk." Inter-connectivity and paradigm logic breed innovation, discovery, and cultural change through paradigm shifts. Finally, Value Sharing "delights the customer" and long-term relationships (Winder, 1993).
- *Registry* - Recognized as service directory. A service registry stores information about services in a SOA environment. At a minimum, the registry includes information that participants can look up and find out the location of the service and what it provides. A registry may also include information about policies that are applied to the service, such as security requirements, quality of service commitments and billing (Erl, 2005).
- *Universal Description, Discovery and Integration (UDDI)* is the registry standard from the Organization for the Advancement of Structured Information Standards (OASIS). UDDI is well supported by large suppliers of software nowadays, the intentions with UDDI is to use it as a catalog to manage web services, the registration will create a universal explanation, which will permit services to be exposed and then integrated together (Abrahams, 2005).

2.2 Theoretical model

This is our own picture of this master thesis theoretical frame of reference, we searched and discovered important building blocks to illustrate and declare in the theory. This model presents different blocks that we believe are a part of a strategy for SOA and importance of control. This model also represents our thinking process that led us to a result. Every box in the model will be enlightened in this chapter, all but OOAD (object oriented analysis and design) which is a fundamental well-known concept of software engineering and will not be declared in this thesis. This model starts with the significant statement of bridging the gap between business and IT, a great deal of the problems we see to day is built on bad communication between these two elements.

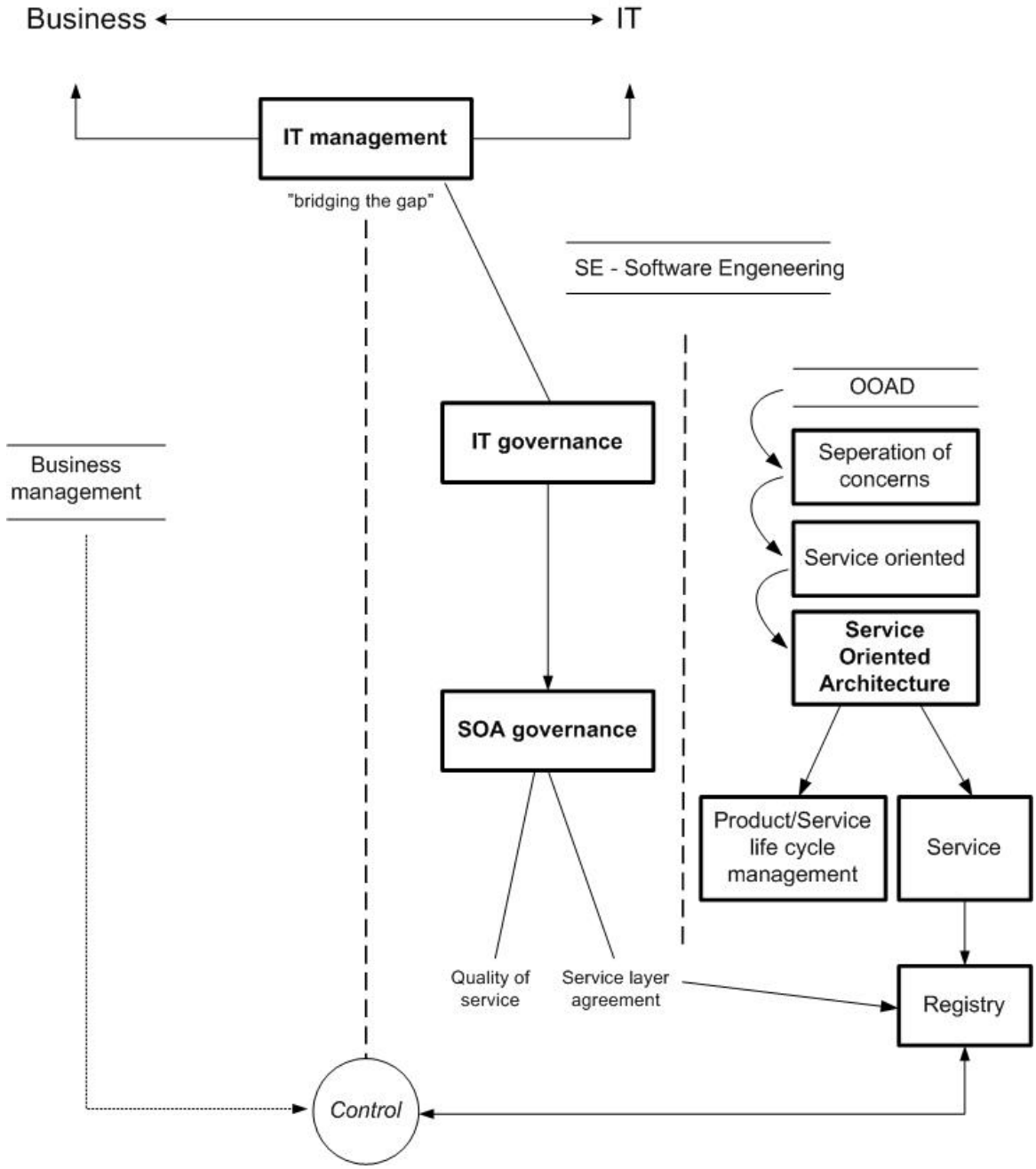


Figure 2. Our theoretical model

2.3 IT management

2.3.1 What is Management?

There are many views concerning what represents "management", and clearly the belief of management as an activity is not new. There is a peculiarity that can be made between management as a 'cadre' of people, and management as an activity. Management as a cadre is those members of an organization who bear the title of manager and who normally in a similar way share beliefs about their position and right to supervise. (Easterby-Smith, Thorpe & Lowe, 2002)

With the growth of the service sector, the move from hierarchical structures to lean empowered organizations, the development of virtual teams and global corporations, the traditional means of defining a manager is becoming increasingly problematic Castells (2000) and Handy (1989, 1996) has noted this change, when he says that in new organizations, everyone must have management skills. It is not just the traditional resources such as people, capital and technology which need to be managed; an increasing emphasis is being placed on the management of intangibles such as quality and knowledge, and many authors e.g. Pralahad and Hamel (1990), Nonaka and Takeuchi (1995) also Moingeon and Edmondson (1997) now believe that these factors hold the secret to competitive advantage.

The modern use of the term 'management' derives from the USA, with the requirement for business and entrepreneurial skills at the turn of the century when American industries and railroads were developing very rapidly (Lawrence, 1986). Two of the leading persons during this period were Taylor (1947) and Fayol (1916/50) who classified the main functions that a manager should perform, such as: planning, organizing, co-coordinating and controlling. Although this classical view of management has much face validity, later researchers showed that these functions had little resemblance to what managers, whether good or bad, actually did in their work (Mintzberg, 1973).

The purpose of IT management is to administer IS/IT activities with the business activities, a valuable definition of IT management is given by Magoulas and Pessi (1998) when they agree that IT management is about dealing with issues such as the use of new technologies, systems development and maintenance, and strategic exploitation of possibilities offered by the technology. It also states that IT management is the art of improvements through design or balance of dependencies among people, among computers and among people and computers.

IT management deals with issues concerning organization of work processes, functions, structures, goals, missions of various stakeholders and its dependencies to IT. Responsibilities and authority, and to handle or avoid risks that evolve and might jeopardize successful management is a part of IT management. Moreover it includes handling problems that deals with applying new technologies, development of new information systems, maintenance of existing investments in IT and in a long range perspective utilize and develop the possibilities that IT offer. This includes new ways of conducting business (Magoulas & Pessi, 1998). Magoulas and Pessi (1998) relate the origin of the term IT management to the increase in environmental dynamics and rapid technology development which resulted in a situation where long range planning was no longer sufficient. Instead a more continuous process of IT-planning was needed, and so the term IT management was coined. Management remains a unique method and it can not be seen as a universal model because management is situational.

2.3.2 Development of IT management and Information environment

The history of the IS/IT era can be divided into at least three epochs: data-management, information-management and the network epoch. These epochs reflect how IT has been used; furthermore what has been seen as relevant management questions (Earl, 1989; Ward et al., 1990; Brancheau, 1997).

Since the 60´s organizations have implemented and used information systems to support the enterprise (Magoulas & Pessi, 1998). Computers where mainly used for issues related to calculation such as accounting and other administrative tasks during the Data-management epoch, 1955´s to mid 70´s (Earl, 1989; Ward et al., 1990; Brancheau, 1997). In the early 60´s there was a vision to concentrate all managing of information and handle all information support within an organization with a totally integrated system (Barlow et al., 1966). IT-personnel worked separate from the rest of the organization and they developed their own culture and terminology. The responsibility for IT was delegated to computer experts who were not normally included in company management (Earl, 1989; Ward et al., 1990; Brancheau, 1997). Moreover, a problem situation for many organizations became more and more critical (Magoulas & Pessi, 1998).

During the 70´s insidious tendency of "anarchy" began, organizations subdirectories took the question by themselves and gained detached information systems (Magoulas & Pessi, 1998). The PC-revolution resulted in an Information- management epoch; the development of more advanced systems for daily work and technical functions, also the relation between price and performance was drastically improved (Earl, 1989; Ward et al., 1990; Brancheau, 1997). The result required extensive manual effort to transmit information between information systems. Furthermore, this created huge problems with inconsistency and redundancy (Magoulas & Pessi, 1998).

IT-managers in many organizations began during the Information- management epoch to approach the company management but the questions usually stayed at cost handling and technical themes. Several corporations became dependent on administrative systems to be able to manage their daily activities. IT-systems influenced many questions outside data-management such as organization of work, economy and inter-organization al co-operation. Management information systems strengthened its position within organization s and information was starting to be viewed as a resource. (Earl, 1989; Ward et al., 1990; Brancheau, 1997)

First generation database, data communications, and real-time on-line systems were new technologies and it was considered critical to keep up with these technologies so that systems did not come crashing down. Managing IS was considered much more than a technical issue, ways to align IS with organizational strategy were needed. Steering committees were created to solve this (Earl, 1989; Ward et al., 1990; Brancheau, 1997). The development of information systems accelerated in an enormous speed, several new application areas were found. Moreover, the significance of integrating information system was noticed (Magoulas & Pessi, 1998).

The network epoch (1985-) is characterized by IT usage not only incorporating data- and information management but also communication and interaction. Internet and virtual organization s are examples of constellations that have grown out of this era. At the same time traditionally separated areas such as IT-usage and technical IT is being increasingly integrated. The cost of IT is more relevant than ever before, since the prime topic for many

companies in this epoch is to cut costs to survive. (Earl, 1989; Ward et al., 1990; Brancheau, 1997)

The role of IS in competitive strategy and in the value chain started to grow in popularity. Information systems were starting to be viewed as a strategic tool. IS also had a role in organizations target on cutting overhead and regaining competitiveness. Leading executives had become more of an active participant in corporate strategy and external affairs. Many IS managers came from a mixed background which included both business and technology. Flatter, team-based, customer driven organization s focusing on business processes and communications with suppliers and customers evolved. Databases contained extensive information about buyers and were seen as a new corporate asset. (Earl, 1989; Ward et al., 1990; Brancheau, 1997)

2.3.3 Management science

Checkland (1981) state that if there was a social science on the pattern of the natural sciences, with assumptions and laws, and a body of theory which tells a story in which the hypothesis and the laws are meaningful and logically linked. Then the social science would certainly be at help in the solution of "real-world problems", just as natural science is available to help technologists and engineers solve their problems. Yet it is not awkward to predict that, an traditional social science would help when a decision taker face problems in social systems. By "real-world problems" Checkland (1981) means problems of decision, problems in social systems, problems which arise and problems which we find ourselves facing, in contrast to the scientist's problem in a laboratory which he/she can define and limit. Checkland (1981) further state that real-world problems are of the kind:

> What should the British Government do about the supersonic passenger aircraft project, Concorde? […] Should I marry this particular girl? Shall I change my career? Such problems are in fact problems of management, broadly defined. The management process, not interpreted in a class sense, is concerned with deciding to do or not do something, with planning, with considering alternatives, with monitoring performance, with collaborating with other people or achieving ends through others; it is the process of taking decisions in social systems in the face of problems which may not be self-generated. This being so, we might expect that so-called ´management science` is in fact body of scientific knowledge and principles relevant to the management process (Checkland, 1981, pp 72).

Checkland (1981) say that management science definitely is present in the sense that there are experts who would think of themselves as ´management scientists`, and the institutions which they belong to have books, journals and conferences dedicated to management science on top of that courses and education. Managers in industry and in the public sector, leaders of trade unions, politicians and administrators etc. have not been convinced that management science is more than slightly related to their task. Checkland (1981) state that most of those managers would agree with Drucker (1974) when he state that: *"Management is a practice rather than a science. It is not knowledge but performance"* (Checkland, 1981, pp 72 quote Drucker, 1974).

As early as 1967 expressed Ackoff about the importance of management, there are five assumptions frequently made by designers of management information systems that can be identified. It is discussed that these are not legitimated in many matters and therefore lead to major absences in the resulting systems. These assumptions expressed by Ackoff (1967) are:

> (1) the critical deficiency under which most managers operate is the lack of relevant information, (2) the manager needs the information he wants, (3) if a manager has the information he need her decision making process will improve, (4) better communication between managers improves organizational performance, and (5) a manager does not have to understand how his information system works, only how to use it (Ackoff, 1967, pp 147).

Most MIS (Management Information Systems) designers are determined what information is needed, and to get MIS information they simply ask managers what information is essential in their work. This is based on the assumption that managers know what information they need and demand. For a manager to know what information she needs they must have an adequate model of each information need and these conditions are seldom satisfied. Most managers have some idea of at least some of the types of decision they must make. Give the manager the information she needs and her decision making will improve, it is indispensable to determine how well managers can use needed information. When, because of the complexity in the decision process, they can not use it well, they should be provided with decision rules or performance feed-back so that they can identify and learn from their mistakes. (Ackoff, 1967)

2.3.4 Chapter summary and highlights

IT management is in fact a concept and a research field of great relevance; it is one of many important presumptions on the topic of SOA and service registry. Since the old way of thinking has led to a complicated IT environment which delays enterprise development rather than support it, IT management is, jointly with other concepts, important to put into practice making IT environments better. It is most significant to take IT management in consideration regarding business integration because IT management support business and application integration projects, and contain handling problems that deals with applying new technologies, expansion of new information systems, maintenance of existing investments in IT and in a long range point of view extend the possibilities that IT offer.

2.4 IT governance

Conventionally, governance for information technology and information systems has been understood and have had focus on IS/IT decision making authorities, or as an executive framework. IS/IT governance from this perspective should ensure expected benefits of IS/IT to be controlled in a desirable manner. However, when the information environment change and become more and more complex and the dynamics with new participants in the organization success, changes in external market and new stakeholders, IS/IT business governance is systematically determining the needs and performance of company IS/IT investments bearing in mind the demands and need of the stakeholders. (Kakabadse, N.K., & Kadabadse, A., 2001)

In particular, decisions related to the initiation of information systems projects, the implementation of these projects and their contribution to overall business strategy are likely to fall under governance inspection responsibility. Within this definition, those involved includes any group or individual that can affect, or is affected, by decisions regarding corporate information technology. Hence, IS/IT governance is a broad concept (Kakabadse, N.K., & Kadabadse, A., 2001). The demand for governance is based on the growing demand for corporate responsibility concerning management and integration of the service payment to

organizational performance, thus creating a rising demand on effective IS/IT systems (Sifonis & Goldberg, 1999). IT governance can be defined in many different ways; according to Grembergen HICSS conference (2006) is the meaning of provided definitions the same. IT governance can be described as the aim to create simplicity about decision rights and accountability, to create an adequate level of desirable behaviors within and among organizations in a complex enterprise. There are a number of useful definitions of IT governance and in this thesis we use the above described.

2.4.1 Definition and meaning

IT Governance can also be explained as the schema for specifying responsibility areas used by organizations to achieve attractive performance in the use of information technology (IT). IT governance should not be used to make specific IT decisions, the management does that. Instead IT governance determines systematically who should make and contribute management decisions (Ross & Weill, 2004). According to the theory of IT governance an enterprise should consider two sides if they are about to adopt IT Governance; *behavioral side* and *normative side*.

Behavioral IT governance focuses on relationships and patterns of behaviors between different participants with different roles and requirements in an organization. Participants include all people that have interest and gain value in the way the organization works, for example managers, shareholders, employees, creditors but also key customers, and communities. A framework of rules is a common way to describe the *normative* side. It refers to the rules and to the structures that implement the relationships, private behaviors and it constructs the corporate strategy. Examples of normative governance regulations can be factors such as company laws, security policies and requirements, both external and internal. (Ross & Weill, 2004)

2.4.2 Important IT governance concepts

Enterprises who want to apply the concept of IT governance have two issues; *what?* and *who?* Those two questions have to do with responsibility spheres and will lead us to five key decisions presented by Ross and Weill (2004) and organizations must clarify and resolve those along the way to format of a properly IT governance strategy. Relationship and linking between these key decisions are important factors for effective governance.

IT principles are guidance instructions and will help the organization to clarify the use of information technology in the business role. These instructions are linked with defining integration and standardization described in the *IT architecture* principles. All principles build on each other and are like an iterative loop were the organization can toss and turn the principles to meet the overall requirements of the enterprise. *IT infrastructure* can assist the enterprise, with help of different methods, to inspect which services they need and provide. In this way the organization can establish which services should be shared external and which should be internal. *Business application needs* are concerned with the business' need for IT applications and software. Which applications should be internally developed and which should be purchased from external vendors, to fit specific requirements? Guidance for *IT investment and prioritization* summarize which initiatives to fund and how much to spend on each. (Ross & Weill, 2004)

Through well defined collaboration spheres, Hugoson (1986) means that information systems become more flexible and decoupled and a clearer mirror of the enterprise will appear. The responsibility for a business process also includes responsibility for the information supply.

Hugoson points out that both information and information systems are very important assets to the organization and they must manage them well to make rapid and accurate decisions. The people who carry out the business within a function or process must have access to information and know that it is *correct*, *available*, *secure* etc. these attributes are related to the concept of information security and information handling and they are a huge part of IT governance (SIG Security et al, 1997). The information access should be settled in a way that supports flexible constellations among partners and the availability depends on the capability between information systems.

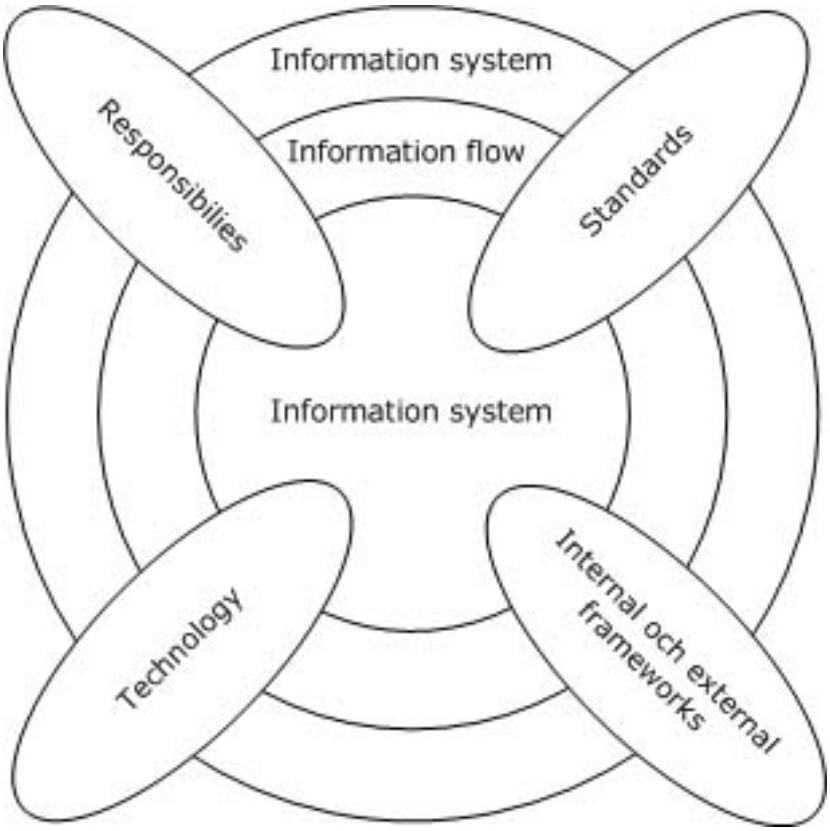


Figure 3. Dependencies in information exchange (SIG Security, 1997)

Ross and Weill (2004) express this in terms as IT principles that drive the architecture that leads to infrastructure. IT governance involves defining who will be responsible for both input and decision making for each decision and it is important to link the strategic business people with the IT people. Do people have the same language for communication and that business an IT meet in the middle for successful IT architecture.

2.4.2.1 Why IT governance is important

Awareness, *concentration* and *management time*, are three key words for effective IT governance. In today's global business world, contemporary enterprises in growth have seen that information is important for competitive advantage. The use of IT and governance can help these enterprises in their activities and for managing information. Creating your decision process based on governance theories, minimizes the risk to make misleading decisions. High-quality IT governance harmonizes decisions with desired behaviors and business objectives. Without carefully designed and implemented governance structures, enterprises

leave this harmony to chance. There are several reasons why IT decision should not be left to chance and thus needs good governance **(**Ross & Weill, 2002**)**. Eight of the reasons follow below:

- Good IT Governance Pays Off, studies have shown that firms with specific IT governance strategies above average had superior profits as measured by a three-year industry adjusted return on assets (ROA). The differences varied by strategy of the firm, but the above-average-governance-performing firms had ROAs more than 20 percent higher than the firms with poorer governance pursuing the same strategy. (Keen quoted by Devaraj & Kohl, 2002)

- Information technology is expensive; the average enterprise's IT investment is now greater than 4.2 percent of annual revenues and still rising (Grolmolski, Grigg, & Potter, 2001). This investment results in IT exceeding 50 percent of the annual total capital investment of many enterprises. As IT has become more important and persistent, senior management teams are more and more challenged to manage and control IT to ensure that value is created. To deal with this issue, many enterprises are creating or refining IT governance structures to focus better on IT expenses and strategic priorities.

- IT is pervasive, according to Keen (2002) well designed IT governance actions help organizations to distribute IT decision making to those responsible for outcomes. It is no longer desirable or possible for many enterprises to centrally manage their IT and information. In early years the requests for IT costs came only from the IT group and no management were involved. Today enterprises are more aware that IT costs must originate from every corner of the enterprise and just not the IT division. Some estimates suggest that only 20 percent of IT expenses are visible in the IT budget (Keen quoted by Devaraj & Kohl, 2002). The rest of the expenses occur in other business reports such as (business) process budgets, product development budgets, and every other type of budget.

- New Information Technologies Bombard Enterprises with New Business Opportunities – Foresight in establishing the right infrastructure at the right time enables rapid implementation of new electronically based business initiatives as well as consolidation and cost reduction of current business processes. Foresight is more likely if an enterprise has formalized governance processes for harmonizing desirable behaviors and IT principles. Enterprises must respond rapidly to the threat of competitors and a flexible IT infrastructure balance dual needs of cost effectiveness and business requirements. Service oriented architecture provides flexibility to support future business needs. The introduction of new technologies give enterprises new opportunities to design their information architectures based on autonomy and flexibility and enterprises are well prepared to strategic threats and take care of opportunities. (Shapiro & Varian, 1999)

- IT governance is critical to organizational learning about IT value and to better understand the value of IT-related initiatives and enterprises must find new ways of present IT spending, not only through traditional discounted cash flow analysis. Effective IT governance creates mechanisms through which enterprises can debate potential value and formalize their learning. Well designed governance strategies also facilitate learning by formalizing exception processes. Enterprises often learn through

exceptions, effective governance makes learning via exceptions explicit and shares any new practices across the enterprise if appropriate. (N.K.Kakabadse and A.Kadabadse, 2001)

- IT value depends on more than good technology; as IT implementations enable increasing standardization and integration of business processes, the roles of technologists and business leaders become progressively more intertwined. When senior managers abdicate to IT executives' responsibility for IT success, disaster often ensues (Ross & Weill, 2002). Successful firms not only make better IT decisions, they also have better IT-decision-making processes. Specifically, successful organizations involve the right people in the process. Having the right people concerned in IT decision making yields both more strategic applications and greater buy-in. These more involved people then produce better implementations. In recent years estimations place IT failure rates at over 70 percent of all IT projects, IT governance strategies help enterprises to minimize this high failure percentage (The Standish Group's 2001 Chaos Report, 2001).

- Senior management has limited bandwidth and with carefully designed IT governance organizations get a clear, transparent IT decision-making process that leads to consistent behavior linked back to the senior management vision while empowering everyone's creativity. (Ross & Weill, 2004)

- Leading enterprises govern IT differently. Present studies have shown that depending on the business goal, market and so on, top-performing firms govern differently to achieve the expected behavior. However all top-performers' governance had one aspect in common; their governance made the tension transparent around IT decisions such as standardization versus innovation. (Ross & Weill, 2004)

2.4.3 Chapter summary and highlights

IT governance engages defining who will be responsible for both input and decision making for each decision. Good IT governance complements decisions about the management and use of IT with business objectives. Especially, decisions related to the start of information systems projects, the implementation of these projects and their contribution to overall business strategy are likely to fall under governance inspection responsibility. IT governance as an alternative determines systematically who should make and supply management decisions. According to the theory of IT governance an organization should think of two sides when adopting IT Governance; behavioral side and normative side.

2.5 SOA Governance

Design rules united with enforcement are typically called *policies*. The development and enforcement of SOA policies and procedures goes by the name SOA governance. Governance and architecture go hand in hand. SOA governance provides context for system architects and designers (Java Sun, 2006) and SOA governance is an extension of IT governance and is used in the context of service oriented architectures. The principles are the same; to create simplicity about decision rights and responsibility. The goal is to create a sufficient level of desirable behaviors (WebLayers, Inc, 2005).

Because of the flexibility and the dynamic structure of SOA, SOA governance has an impact on existing corporate IT governance processes. IT governance and SOA governance are

linked together and some of the governance processes in IT governance are involved in SOA governance, e.g. budgeting and project approval process, portfolio management activities, and ongoing oversight of projects to assure budgetary fulfillment and integration issues. Aligning governance for SOA activities is critical because there may have to be changes in the normal IT governance processes for budgeting and portfolio management. (Carlson & Marks, 2006)

Windley (2006) say if there is no control of projects or no one that influence people to adopt SOA and service reusability as a fundamental part of design, then projects will go their own way, driven by requirements of a particular business unit or project. The same goes for the portfolio management process; if there is no mechanism to surface SOA and no opportunities or control concerning project reuse then there will be budgetary pressure toward an SOA, because projects will similarly go their own way. SOA governance is a way to influence behaviors of business units, as well as IT and business personnel within them, to more aggressively support SOA and reuse.

If you think about the process of information architecture as two hands; In SOA Web Service Journal by Carlson and Marks (2006) in the first hand we have strategy and goals, followed by definition of elements, standards, and organization of architecture to accomplish those goals. The second hand is the application architecture to fund projects, the acquisition or implementation of various technologies and standards, and the enforcement of compliance to the enterprise architecture goals (Carlson & Marks, 2006). Enterprise architecture processes may undergo changes initiating an SOA implementation in an organization. The architecture process and the organization will have to be restructured to accommodate the requirements of an SOA because the skills, roles, and functions of an enterprise architecture team are not completely appropriate for an SOA initiative.

2.5.1 Best Practices for Good Governance

How can we integrate IT governance with SOA governance without losing control of our existing projects? Because of the loosely coupled connections between objects within SOA and the constant evolution of the information, which architectures are based upon, SOA governance has implications on existing corporate and IT institutions as well as new organizational structures and processes.

Focus must be on comprehension on what governance is and what it means, how the organization best achieve an adequate level of governance and how the governance strategy will be implemented. It also concerns how and why governance is necessary and how it creates valuable in the SOA migration. (Carlson & Marks, 2006)

Windley (2006) manifests that SOA governance is social in nature, inviting continuous dialog between developers and architects, furthermore Windley call attention to some guiding principles to more easily integrate SOA governance to existing IS/IT governance strategy.

- Broad review – enterprises should deploy policies for governance, especially in the context of SOA so that development, maintenance, and modifications become efficient. The organization should ensure that all employees know what and when they are allowed to do things. People, who know what is happening on the “ground” must have real input and together they will decree, determine, regulate and recommend the policies

- Framework for interoperability – as described earlier SOA is based on open standards. Organizations should begin with building an extensible interoperability framework that details the protocols used in business processes and agree on standards.
- Granularity – policies must not be overly detailed. Over detailed policies are difficult to preserve and confine creativity. Provide plenty of flexibility for services and applications with low risk should be developed.
- Communication – if policies become heavy and complex they will be difficult to understand and comply, which leads to extensive explanation. Managers must ensure that updates are properly distributed, soliciting feedback along the way. Larger corporations should establish Centers Of Excellence (COE) with full-time staff dedicated to supporting SOA, including governance. An effective COE provides the guidance and education that holds your governance effort together.
- Meaningfulness – without enforcement, policies mean little. Services should be designed and developed with policies in mind and that employees really follow the policies and sees them useful. Designers, developers etc. have to be sure that they truly know the consequence of violations. If an organization already got a strong IT governance process in place, it will supply them well as groundwork for SOA governance.

Roman Stanek chief software architect and founder of the SOA registry supplier ´Sysnet` is quoted by Windely (2006) and Stanek says: *"Without SOA governance, you end up in a Web services version of DLL hell. SOA governance gives consistency, predictability, and allows big applications to be built from small pieces"* (Windley, 2006, quote Roman Stanek, January 19, article 73698, 2006-06-08). Anne Thomas Manes, vice president and research director at Burton Group is quoted by Windely (2006) and agree with Roman Stanek when she says; *"Most organizations will fail miserably if they don't implement the right form of governance. SOA is about behavior, not something you build or buy, you have to change behavior to make it effective"* (Windley, 2006, quote Anne Thomas Manes, January 19, article 73698, 2006-06-08).

Those quotations emphasize the importance of the two concepts IT governance and SOA governance which enterprises have to manage in a complex global networked environment with many different services, actors and behaviors. The decision-making process can be *organized* in various ways, but ultimately it is a social process that has to be merged into your organizational culture. SOA governance incorporates best practices around *organizational dynamics* and how human beings behave in organizations. Simply put, governance processes should make it easy to do things the right way and hard to do them the wrong way.

Polices can affect every aspect of the service lifecycle, including design, deployment, and operation. In most organizations, it makes sense to begin policy-making efforts with standards. After all, standards make SOA possible. Each enterprise must determine which standards are used where and when. You could call out specific standards and individual policies, but a better strategy is to create an interoperability framework (IF). An interoperability framework is an special policy that lists the standards that your organization will use, points to reference information, and indicates the status of the choice: approved, de facto, emerging, sustained, sunset, or in process. (Windley, 2006)

Interoperability framework separates references to quickly changing standards from individual policies, making them easier to manage. Enterprises should look for systems that not only allow them to manage, version, and discover policy at design and run time but also

provide for policy reuse. Being able to create and manage policies independent of services allows you to fully leverage policy assets. In general, enterprises should automate as much of the governance as possible. That requires a centralized investment in people, organizations, and tools to establish the appropriate context for SOA. Properly done, your governance process and its associated infrastructure may be the only centralized elements in your entire SOA deployment. (Windley, 2006)

2.5.2 Chapter summary and highlights

The enlargement and enforcement of SOA policies and actions goes by the name SOA governance. Because of the flexibility and the dynamic structure of SOA, SOA governance has an impact on existing corporate IT governance processes. SOA governance provides context for system architects and designers and SOA governance is an extension of IT governance and is used in the context of service oriented architecture, the principles are the same. Service oriented architecture requires a major shift in the way software is developed and deployed within enterprises. Organizations want to ensure continuity of business operations, manage security exposure, align technology implementation with business requirements, manage liabilities and dependencies, and reduce the cost of operations. With the new paradigm, technologies, and standards companies need to implement their SOA in a well planned, well coordinated, and effectively managed way. Governance and architecture go hand in hand. The need for SOA governance is business-oriented.

2.6 Separation of concerns

According to Jazayeri et al. (2000) the term *separation of concerns* was coined in the early 1970s by Dijkstra (1974) and Parnas (1972) and refers to the importance of dealing with one significant issue at a time. To keep away from program code which deals with many issues simultaneously, programming tries to separate each issue into a distinct set of code. These pieces of code are then combined to generate a needed component. The idea is borrowed from Aspect-Oriented Programming (Jazayeri et al., 2000).

Separation of concerns is a general principle in software engineering as an answer to control the complexity of ever growing programs. Separation of concerns means that every employee has access to information which is needed to attend to do his or hers work; the employee has access to do the work he or she has authority to. Actually, a large amount of information is handled in confidential to minimize the risks. The system administrator should for example not have access to sensitive information about future budget closing (Simonsson, 2005). In Software Engineering, Separation of Concerns is the process of breaking down a program into distinct components that perform distinct tasks. The purpose and benefits of such modular approaches include reduction of complexity, ease of updates and changes, reuse, simplifying customization (Ossher et al., 1999).

Ciccarese et al (2005) state that Separation of Concerns is a general approach that can be applied to human resources, demonstrating that each individual exhibits are different concerns that are usually associated with his/her skills, attitude or past experience. Not everyone is alike, not everyone achieves the same job with the same capability. Moreover, it can be observed that dividing people with common skills in different working groups increases productivity and reduces management costs.

Concerns are the primary motivation for organizing and decomposing software into manageable and comprehensible parts. Many kinds of concerns may be relevant to different

developers in different roles, or at different stages of the software lifecycle. For example, the prevalent concern in object-oriented programming is the class, which encapsulates data concerns. Feature concerns, like printing, persistence, and display capabilities, are also common, as are concerns like aspects, roles, variants, and configurations. Appropriate separation of concerns has been hypothesized to reduce software complexity and improve comprehensibility; promote traceability; facilitate reuse, non-invasive adaptation, customization, and evolution; and simplify component integration. (Ossher et al., 1999)

To benefit fully from separation of concerns, one must have the right modularization at the right time: the concerns that are separated must match the concerns one need to treat. Unfortunately, different development activities often involve concerns of dramatically different kinds. For example, changing a data representation in an object-oriented system might involve a single class, or a few closely-related classes, and might be done additively using sub-classing or suitable design patterns. (Ossher et al., 1999)

Ossher et al (1999) use the term multi-dimensional separation of concerns (MDSOC) to refer to bendable and incremental separation, modularization, and integration of software artifacts based on any number of concerns. MDSOC promotes reuse, improves comprehension, reduces the impact of change, eases maintenance and growth, improves traceability, and opens the door to system re-factoring and reengineering. MDSOC promotes reuse, improves comprehension, reduces the impact of change, eases maintenance and growth, improves traceability, and opens the door to system re-factoring and reengineering.

According to Morris (2006) separation of concerns has been with us for a long time. For example, the way networks are designed and built: devices are selected, configured to run usually several protocols, and services are delivered across the network.

Service-orientation is said to have its roots in separation of concerns, it has been implemented in different ways with different development platforms. Object-oriented programming and component-based programming approaches, for example, achieve a separation of concerns through the use of objects, classes, and components. Service-orientation can be viewed as a distinct manner in which to realize a separation of concerns. The principles of service-orientation provide a means of supporting separation of concerns while achieving a foundation paradigm upon which many contemporary SOA characteristics can be built. (Erl, 2005)

2.6.1 Chapter summary and highlights

One approach is to apply the idea of separation of concerns to software evolution. In that context, separation of concerns allows to separate parts of the software that exhibit different rates of change or different types of change. This makes software evolution more manageable. Service-orientation can be viewed as a distinct manner in which to realize a separation of concerns. Separations of concerns also permeate society. For example, if you want a bank loan, it is unlikely you would make much progress by going to the dentist. Rather, you go to the service provider who can deliver your requirements.

2.7 Service orientation

According to Kingstedt (2005) the concept of Service Orientation has different meaning depending on the employee's role in the company. The principles of service orientation provides significance to the technicians in terms of supporting run-time and selection of

capability providers; This group sees service orientation as a flexible option that allows applications to be sensitive to the content and context of a specific business process, and to gracefully incorporate new capability providers over time. It is a way to create dynamic, cooperating and loosely coupled applications.

To the business people and management such as CEO[1], CFO[2], CIO[3]; service orientation supplies a model for collecting information and business logic of multiple systems into a single interface. Service orientation allows diverse and redundant systems to be addressed through a common, coherent set of interfaces. To the responsible for the enterprise, the Managing Director (MD) service orientation means encapsulation of business applications behind capability based interfaces, the service model allows controlled access to mission critical applications, and creates the opportunity for continuous improvement of the implementation behind that interface. Service orientation protects investments from the swirl of change. (Kingstedt, 2005)

Last but not at least, service orientation means to the enterprise and business analyst that he/she can describe and mapping employees, external capability providers, and automation systems into a single model. The analyst can better understand the cost tradeoffs associated with investments in people, systems, and sourcing. (Kingstedt, 2005)

2.7.1 Service orientation modeling principles

Service-orientation has become a distinct design approach which introduces commonly accepted principles that govern the positioning and design of our informational components. When service-oriented business logic is comprised together we get what we refer to as a *service-oriented solution.*

- When modeling and designing services based on service oriented principles; a service should fulfill a number of characteristics; (Erl, 2005). A service should;

- Advocate relationship between services that are loosely coupled and minimized upon dependencies. A service requires only awareness of each other.

- Base its communication on agreements with other services and a service should own its own service contract. Communication rules are defined by one or more service descriptions and related documents that belong to the service.

- Fulfill the autonomy characteristic and the processing logic services encapsulate belongs to the service defined and to no other. Services which are interested in another service may revoke it in one way or another, but does not need to know how it is implemented and in what way.

- Hide their logic from the outside world and be abstract. As described above, communication between services are based on service contracts, not logic.

- Divide the service logic which they contain into autonomous services with the intention of reuse and maintenance.

[1] Chief Executive Officer
[2] Chief Financial Officer
[3] Chief Information Officer

- Be compassable and a collection of services or a single service can be coordinated and assembled to form composite services. A particular service can in fact be the sum of several other services.

- Support statelessness, the information about a specific activity does not require any data storage. Activity data should be kept at a minimum level.

- Up hold discoverability and service descriptions should be composed and designed so that they easily can be found and assessed via available discovery mechanisms.

2.8 Service

To fully understand the concept of Service Oriented Architecture and to effectively describe it the two concepts Service and Web Service has to be cleared out.

2.8.1 What is a service?

Services are according to the economic science and marketing science the non-material equivalent of physical goods. There is no ownership regarding services and services are mostly related to an economic action involving consumer and supplier. The ownership is the prior difference between providing physical goods and providing a service, also human behaviors and their physical possessions tend to change when consuming a service, it might be a hairdresser cutting a customers hair in exchange of money. Services is claimed to be the process that creates benefits for both consumer and provider by changing the state in human assets. Services do not only exist between man-to-man, man-to-service, but services can also be consume information from other services e.g. service-to-service interaction, like the cost free business driven customer service Pricerunner[4] provides. This service involves man-to-service and service-to-service interaction and is a service that provides product- and price comparison to conceivable consumers by searching the Internet for retailers providing the lowest price concerning both products and services. (Magoulas, 2006; Wikipedia 2006)

Supplementary explanation and meaning of services is to provide a number of operations that are of concern to persons who need them, e.g. operations needed in employees' day-to-day work e.g. get costumer details, or a financial transaction over the Internet done by an internet user. So called informational services should contain detailed metadata concerning services and/or collections of services. Further, there are transactional services, most frequently services that do not mainly care of provision of information, e.g. photocopying, printing or banking services. As described previous services can both be material or released on-line as IT services enabled as Web services. (Baker et al, 2003)

Usually organizations identify services as those processes, operations and functions that can create value to the enterprise and fulfill the overall business goal and objectives. As a result, the identified processes can be separated into sub processes such as different business functions, business transactions and system activities. The processes are then analyzed through organizational parameters, guidelines, rules etc. and interpreted into organizational IT services. To make the picture a little bit clearer one can describe business functions in terms of information requests. It could be a clerk asking for information from the system to receive information about customers. At first sight one might think that business transactions consist

[4][http://www.pricerunner.com] Pricerunner provides consumers with easy online access to specifications and prices for a wide variety of products; they do not sell any products by themselves.

of uncomplicated activities that belong to the application which invoked the transaction but in fact business transactions are aggregated activities camouflaged by their own context. System activities are widespread services that can be moved to any platform such as Windows, Linux, Mac-OS-X or UNIX. (Channabasavaiah, Holley, Tuggle Jr, 2003)

Services that interact with each other in changing environments characterized by SOA should complete some characteristics and parameters presented among many authors. Erl (2005) and Sundbland (2004) present common attributes concerning services and a summary of the most important parameters are presented below. A service should;

- preferably fulfill the autonomous demand
- interact through a message based interface
- possess their logic, protect and take care of their own data
- fulfill design rules to achieve loose coupling and be delimited from the outside world
- not share class like objects in OOP[5] but instead they share service schema.

SOA use the same basic generalization that is accepted for both SOAP[6] and XML[7] but is not dependent on them and the idea with SOA is to coordinate IT systems with services that can be called by a service consumer when needed(another service, application etc.) through an interface based on standardized messages (Sundblad, 2004).

2.8.2 Web services

Just like the definition of SOA can be described in many different ways, so can Web services. In this thesis we use Cordeiro & Carvalho (2002) description and definition of Web services;

> Web services are a set of technologies and standards for interoperability well adapted to exploit the characteristics and potentials of the Web environment. The notion of service is of special importance in this context: it refers to a network addressable software component that can receive a call and provide a given behavior on behalf of the calling system, application or component. (Cordeiro & Carvalho, 2002, pp.46)

> A Web service can, thus be defined as an application-oriented service available on the Internet using a standardized messaging system that is not tied to any particular operating system or programming language, since it is conveyed using XML and HTTP. Such messaging systems make portions of code, needed to exchange information between applications, easily readable and reusable by any other system, as they are transmitted in a universally standardized and independent language. (Cordeiro & Carvalho, 2002, pp.47)

2.8.3 Further definition of Services

The people at different levels in the organization have their definition of what a service is. A common way to understand services at operational perspective is to describe them as Information Technology (IT) assets that correspond to real-world activities that can be reached based on policies and governance strategies. Service guiding principles from this

[5] Object Oriented Programming
[6] Simple Object Access Protocol
[7] EXtensible Markup Language

point of view defines who is authorized to access specific services, the performance and reliability levels of the service, and the security levels of the service. People at technical perspective usually sees services as coarse-grained, reusable IT assets with well-defined interfaces that separates services. This separation of conveys in decoupling between service requesters and service providers, enabling both requesters and providers to evolve independently as long as the interfaces remain unchanged.
According to Tomas Erl (2005);

> Service-oriented architecture (SOA) encourages individual units of logic to exist autonomously yet not isolated from each other. Units of logic are still required to conform to a set of principles that allow them to evolve independently, while still maintaining a sufficient amount of commonality and standardization. Within SOA, these units of logic are known as *services*. (Erl, 2005, pp.33)

All of the organizations services must encapsulate information and logic about the environment which they live in to remain independent. The service environment can be specific business tasks, a business entity, or some other logical grouping within the organization (Erl, 2005).

2.8.3.1 Service Identification

The process to identify services within a business or a business unit consists of a combination of top-down, bottom-up, and middle-out techniques. The techniques originate from separation of concerns and these methods can help organizations identify and handle domain decomposition, existing asset analysis, and goal-service modeling. Where to begin in the identification process is dependent on the role the employee has in the overall SOA work and the position in the organization. If one work at business level the top-down perspective is most likely where to begin. Top-down perspective involves business use cases and business process modeling to identify and specify business services. Today there exist a number of tools for business process identification to make the labor easier. Employees' working at technical level often use the bottom-up perspective and tries to identify which systems, applications, hardware etc exists. The identified assets are examined and picked out as feasible nominees to increase cost through implementing them as of underlying services. The middle-out position corresponds to service modeling of those services that not have been identified in the previous two perspectives. The middle-out perspective acts as a venue where top-down and bottom-up get together and discuss there results. (Arsanjani, 2004)

Within SOA, services can and should be used by other services or other programs to make it flexible and dynamic. Services, in SOA and other open systems need in analogy with Hewitt (1987) to have knowledge of its self in order to complete its purpose and effectively understand its own capabilities, knowledge and power. When knowledge is added to a service during its life time, it must transmit and add this new knowledge to its existing knowledge, to make the description of the service updated for successful identification. Services must interact with other services to accomplish something meaningful, i.e. they must exchange data and/or information about their existence and tasks. A communication framework capable of preserving services loosely coupled relationship is therefore required. One such theoretical account is messaging. (Erl, 2005)

2.8.3.2 Service descriptions

Regardless, relationship between services is based on understanding and communication for services to interact and they must be aware of each other for successful interaction. This

awareness is achieved through the use of *service descriptions*. A service description in its most basic format only describes the name of the service and the data expected and returned by the service. The manner in which services use service descriptions results in a relationship classified as loosely coupled. According to Hugoson (1986) is one of the most important qualities of open systems the degree of independence or as Erl (2005) call it; level of loose coupling. Hugoson (1986) discern four types of independence between systems and subsystems, they are as follows:

- Liberty to change data/information within an internal subsystem includes add, change and remove information. When information within a subsystem changes condition, other external systems should not be affected.
- Changed relations between systems and subsystems should be limited to the integrated systems and includes the liberty to change and rebuild the organization al information architecture without existing system are affected.
- That technical connection should be independent, means that the communication system should not know or be influenced when information in subsystems is changed. The communication system should only exist to transport information and data between systems.
- Independence in terms of time includes managing of information whenever the system wants, independence regarding system maintenance; the system should work even if other systems are down.

This part of SOA provides the key ingredient to establishing a consistently loosely coupled form of communication between services, weather they are implemented as Web services or not. For this purpose, description documents are required to accompany any service wanting to act as an ultimate receiver. The primary service description document today is the WSDL[8] definition. A service description includes a point of contract for the provider which provides a definition of the endpoint interface and also establishes the *physical location* (address) of the service. The service description document consists of two spheres that need to be settled and carried out;

- *Abstract description,* establishes the interface characteristics of the (Web) service without reference to the technology used. The abstract description is connected to a;
- *Concrete description* which establishes the transport and location information i.e. attributes for communication, which describes the implemented technology used to execute a service (Erl, 2005).

2.8.3.3 Metadata and service contracts

Transactions and connections between systems create dependencies and each system has to know what they are allowed to do or not. The dispatcher and the receiver in a SOA do not need to know each of the other systems or services data structures, the technology behind the services. Services are not allowed to update other systems that they have contact with unless there are well designed agreements between them.

Hugoson (1986) presents the term *Transaction collaboration* as the transference of one or more objects e.g. a file, a contract between two information systems etc. In relation to transactions we can point out responsibility areas and it is the dispatcher's responsibility for sending information and the receiver has responsibility for the reception of it. The abstract

[8] Web Service Description Language

and concrete descriptions express technical information as to how a service can be interfaced with and what type of data exchange it supports. WSDL frequently rely on XSD[9] schemas to formalize the structure of incoming and outgoing messages. Another common supplement service description document is a policy. Policies can provide rules, preferences and processing details above and beyond what is expressed through the above described descriptions (WSDL and XSD schema documents). Now we have up to three documents that each describes an aspect of a service:

- WSDL definition
- XSD schema
- Policy

These description documents contain *service metadata* or *service master data* and provide essential information's about each service for easier identification within the organizational context. Service description documents institutes the so called *service contract* and a number of parameters should be fulfilled and accepted by requestors for winning communication between services (Erl, 2005).

2.8.3.4 Service advertisement and discovery

In SOA and other open systems there exist no global objects. The only thing that all the various subsystems and services have in common is the power to communicate with each other. Hewitt and de Jong (1982) believe that the most important parameter in open systems e.g. SOA will be negotiation and communication between objects, e.g. services. Similar services and capabilities will arise and evolve in many different locations in open systems. For example different banks can offer similar services that differ in detail from services of its competitors the systems will need to negotiate terms and conditions regarding their transactions. Communications will therefore be a fundamental part and there must be an uncomplicated language in common to promote successful negotiation between partners (Hewitt & de Jong, 1982).

Erl (2005) state that the only thing invoking services need to know about the service they want to contact is the information about the other service's description. When services increase within and outside organizations, they need some form of promote; discover and access mechanisms concerning services descriptions. One way trying to solve the problem concerning an increasing number of services which will avoid service anarchy with organizational services is to apply central catalogues/registries keep track services. These registries can be designed so that both humans and even services can get access to available services. It is essential that the latest versions of service descriptions are available and that new services fulfill critical requirements.

In the beginning of the development concerning a common standard for Web services, finding and marketing of services was a fundamental piece and this is why UDDI[10] emerged and used as a part of early stages of Web service standards. Today UDDI is not commonly applied and it supplies a model for discovery and storage of services. UDDI is today insufficient to fulfill all the requirements that exist in the organization (see section 2.13.1.1) (Erl, 2005). To really benefit from the advantages of a service registry and the advertisement and discovery, a precise definition of services and all to their requirements are necessary. If the organization

[9] The XML Schema language is also referred to as XML Schema Definition
[10] Universal Description, Discovery and Integration, protocol for publication of web services

does not have clear definitions and a common standardized way to define master data concerning services divisions will advertise and define in different ways. Organizations must specify attributes and requirements from the service provider and from the service requester perspective too get the best fundament for advertisement and discovery (Cordeiro & de Carvalho, 2002; Erl, 2005). It is necessary to identify non-functional and functional attributes through different parameters defined by the responsible people involved.

Service provider perspective – the model has to define *who* offers the service i.e. the service provider, *what* the service performs *(functional description)* and on *which channels* the service is available. Moreover, the service could be simple or composed by a set of services organized to a composition framework (Arkin, et al., 2002). Functional description – specifies the operational aspects in terms of invocable *operations* and exchanged *input/output parameters,* which can be specified in a WSDL specification. (Christensen et al., 2001) The *behavior* of a service specifies through identifying which operations that invokes the service. It is important to note that such a functional description relies only on the elements the provider identifies as 'exportable' or, in other words, on the elements that should be made visible for a potential user to know which way he can invoke the service.

The model for *service requests* concentrates on *user* definition (Sloan, 1997). Erl (2005) agrees and say that the requestor of a service is characterized by its profile and by the contexts in which the user can be and in which the user, in a given moment, operates. The requested service is defined in terms of functional and non-functional aspects, as defined in the service provisioning perspective. The profile has a static and dynamic component. The static profile is defined by means of a set of user preferences such as role, which identifies the role played by the consumer while using the application, its Expertise and Ability on the application and a set of Generic Preferences to add application-specific characterizations to the profile, such as device, network, network interface and application protocol.

In particular the user can characterize the requested service through only a subset of the quality parameters suggested by the service community considering only the interesting aspects. (Arkin, et al., 2002)

2.8.4 Chapter summary and highlights

To the business people and management service orientation supplies a model for collecting information and business logic of multiple systems into a single interface. Service orientation allows diverse and redundant systems to be addressed through a common, coherent set of interfaces. Service orientation protects investments from the spin of change. Services are all of the organizational activities and processes an enterprise can identify and moreover divide into sub processes such as business activities, business transactions and system activities. Services work together in a network of software components that can take delivery of a call and provide a given performance on behalf of the calling system, application or component. It is of great relevance to particularly note the difference between services. From an application perspective business activities (processes) are non system activities that are autonomous in their identity.

This part of SOA provides the key ingredient to establishing a consistently loosely coupled form of communication between services implemented as Web services. For this purpose, description documents are required to accompany any service wanting to act as an ultimate receiver. Web services are a set of technologies and standards for interoperability well

adapted to exploit the characteristics and potentials of the Web environment. Web services were actually built to warrant a registry

2.9 Service oriented architecture (SOA)

Describing the nature and identify performed and future actions in environment characterized as open systems in a basic fundament of information/computer system development. Hewitt and de Jong (1983) anticipated in the early 80th that communication will be a greater part of future systems and relationships among systems must be identified because the applications and systems have been developed separately and independently and they will be spread all over the world . Communication is the significant factor that connects these open systems and there most be standard form of communication between the systems. This is essential because often organizations, vendor's etc. has different methods, tools, teams procedures etc. when they develop their systems. Service oriented solutions like SOA are bottomless and under continuous change, this because of the changing environment that the architecture exists in, e.g. organizations will discover new markets and new customers and because of the new requirements they rapidly need to change their way of doing business. New services will be added incrementally and services will be deleted. The single factor that service elements of a SOA and other open systems need to have is the power to communicate with each other in a common language (Hewitt & de Jong 1983).

Service oriented Architectures or service oriented systems can more or less be seen as open systems and the definition of "SOA" is discussed in both journals such as Gartner's "*Introduction to Service-Oriented Architecture*" (2003), and books like "*Service Oriented Architecture: Concepts, Technology and Design*" by Tomas Erl (2005). Essentially, SOA is information architecture based on services enabled through software that starts with a surface definition that forms the boundary between consumer and provider and constructs the entire application topology as a topology of interfaces, interface implementations and interface calls. Yefim (2003) add and suggests that SOA would be better-named "interface-oriented architecture." The fundamental part in service oriented architecture is the set of associations between services and service consumers. Services can be (see Service) huge software components corresponding to a complete business operation or a single business process. Furthermore services are modules based on the organizations, customers, stakeholders and other actors need and demand, and are often in contemporary days attained by its name through an interface, typically in a request-reply mode. When the information environment increases in size, Hewitt and de Jong (1982, 1983) and Erl (2005) constitutes that it will be challenging to access critical information and needed services based on just name, thus must additional attributes be involved in finding of services in the changing architecture that defines SOA.

Modeling information, services and resources become crucial and a fundamental part of SOA and open systems to manage, control, define and discover services. Hewitt and de Jong (1983) speak of the actor system which aims to provide a clean way of implement effects. They define an effect as a local state change in a shared actor, which in their turn changes the behavior in other actors. For example sending a deposit to an account shared by multiple users should have the effect of increasing the balance in the account.

A Service Oriented Architecture is a system; architecture built as separated services that through communication and behavior share information to fulfill the purpose of the enterprise. Services involved in business tasks are a reflection of identified enterprise key processes, translated into IT services with the help of standards to function in the information

environment. Each service interacts with various other services to fulfill an all-embracing task. As the picture below illustrates can one service be a combination of several other functions and in this way uphold the principle of statelessness in the definition of service-orientation. (Erl, 2005)

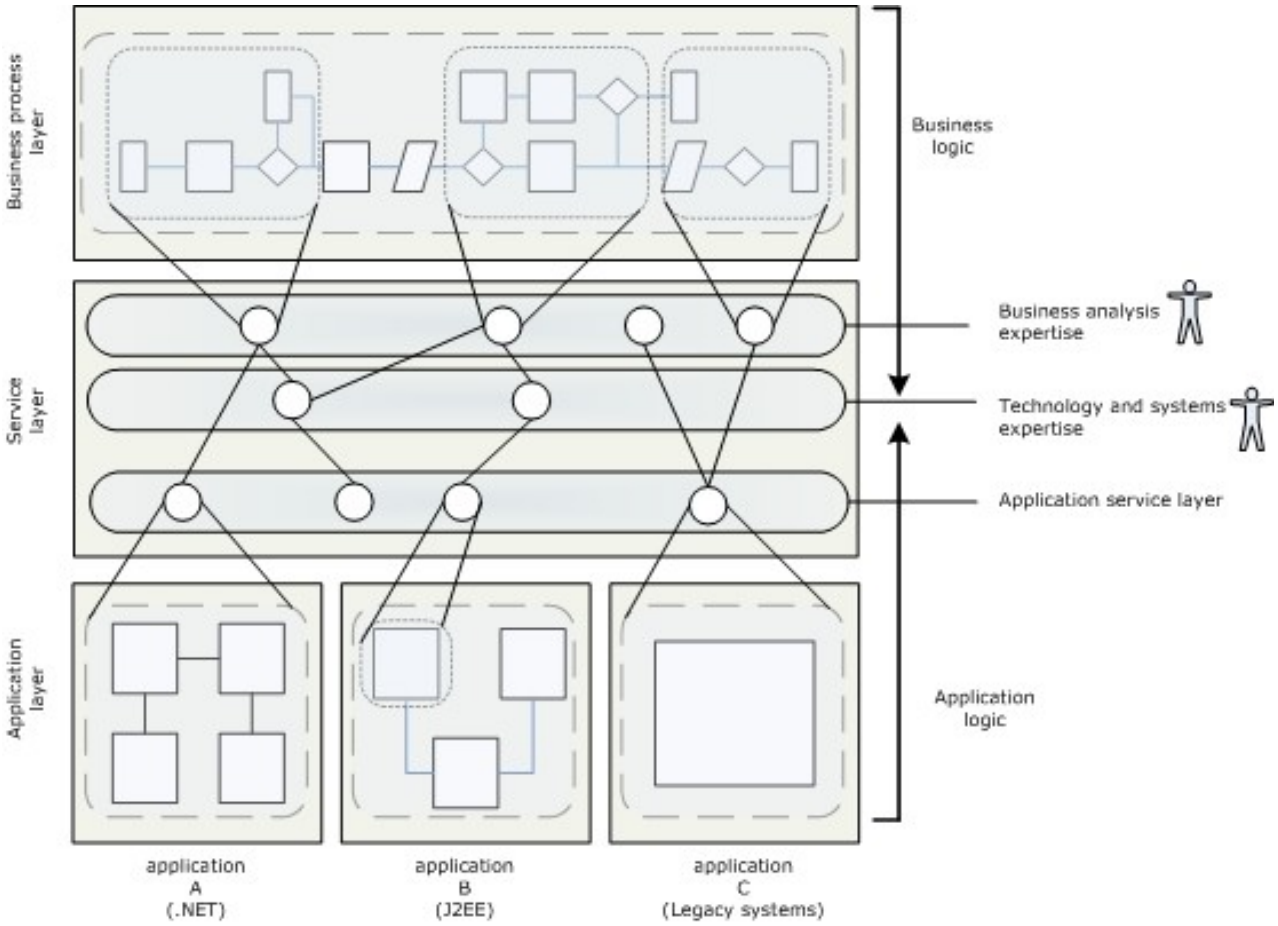


Figure 4. The interface of a Service Oriented Architecture - Separation of business and IT (Erl, 2005)

SOA has many pronounced advantages and benefits (Chatarji, 2004; Erl, 2005; Duberley & Johnson, 2003; Nickull, 2005; Schmeizer, 2005; webMethods, 2006; Yefim, 2006) in difference to conventionally systems such as ERP-systems, but the advantages and benefits may differ from organization to organization, but there are some common advantages;

- Integration will be improved
- Intrinsic interoperability and inherent reuse
- Architectures and software solutions are moving to streamlined development
- Legacy investments are not in vain, legacy systems are fundamental part of SOA.
- Data representations are presented in standardized XML format.
- Organizational focus on investments for communications and other infrastructure
- “Best-of-bread” alternatives
- Organizational agility

As described above, Service Oriented Architecture has numerous advantages, but these benefits do not certify themselves. According to Erl (2005), enterprises should follow the principles of service-orientation and make them established within the enterprise to fully obtain return on investment (ROI). Because of the basic generalization or service-orientation organizations can gain few short-term benefits or advantages. Weather an enterprise achieves those benefits or not, almost certainly has more to do with the organizations information policies and procedures than the code building up the applications.
Service Oriented Architecture demands more organizational discipline than previous development models because of the flexibility and the loose coupling that SOA provides. The flexibility stretches from vendor independent component purchase to integration of systems developed in different computer languages. The huge difference between former development models such as object oriented programming (OOP) and SOA is that service oriented architecture is based upon open standards such as XML. That sort of standardization supplies the foundation for SOA across an organization. To prevent organizations and IT from being overwhelmed by this new complexity, an organization need methods that make serious choices about requirements, design, implementation and maintenance which results in rules to increase control, overview and awareness (Erl, 2005).

The computer industry has developed software solutions to investigate the problem with anarchy concerning services. These software solutions called registries, repositories; catalogues etc. are designed to maintain all rules and under have them under control. But applying a Service Oriented Architecture and demands more than just using SOA tools purchased from vendors, it might demand rethinking in the way the organization do business and changing in employees' behaviors. (Java Sun, 2006)

2.9.1 Chapter summary and highlights

Service oriented architectures or service oriented systems can roughly be seen as open systems and the classification of "SOA" is often argued. Service orientated architecture change the relationships created among systems to open up new possibilities with complex applications, which control services and service orientation to give a single point of usage into multiple systems of records. The introduction of effective service management makes it needed to focus more on an approach guided by the business process and therefore less on functions and components. Modeling information, services and resources become crucial and a fundamental part of SOA and open systems to manage, control, define and discover services. Services implicated in a definite business task are a mirror image of identified enterprise key processes, translated to (IT) services; these services within connected systems work together. Service Oriented Architecture demands more control than earlier development models for the reason of the flexibility and the loose coupling that SOA provides.

2.10 Service management

In open systems it becomes very difficult to determine what objects exist at any point of time (Hewitt and de Jong, 1982). In Brodies and Schidt's; *Perspective on Conceptual Modeling (1982),* Reiter illustrates this with an example of a query that might never finish looking for possible answers. The query could be a conceivable buyer of a used TV and he/she wants to find one in good quality and the information about the object, in this case the TV, can be stored in any of a number of personal and organizational computers. Enormous amounts of effort and time could be expended processing the query to find such a refrigerator, without being certain that the best buy has been located. Similarly if a system is asked to find all the telephone numbers of passengers who have ever made reservations on Pan American, it might

have a hard time answering. It can give all the telephone numbers it has found so far, but there is no guarantee that another one can't be found by more diligent search. These examples illustrate how the "closed world assumption" is intrinsically contrary to the nature of "Open Systems". To understand the "closed world assumption" one can think of the information about the world being modeled as complete in the sense that *all and only* the relationships that can possibly hold among objects are those implied by the given information at hand (Hewitt and de Jong, 1983, Reiter, 1982)

Systems based on the "closed world assumption" are characterized by idea that they can find all the instances of a concept that exist by searching their local storage. In contrast we desire that subsystems be accountable for having evidence for their beliefs and be explicitly aware of the limits of their knowledge. At first glance it might seem that the closed world assumption, almost universal in the A.J. and database literature, is smart because it provides a ready default answer for any query. Unfortunately the default answers provided become less realistic as the Open System increases in size (Hewitt & de Jong, 1982).

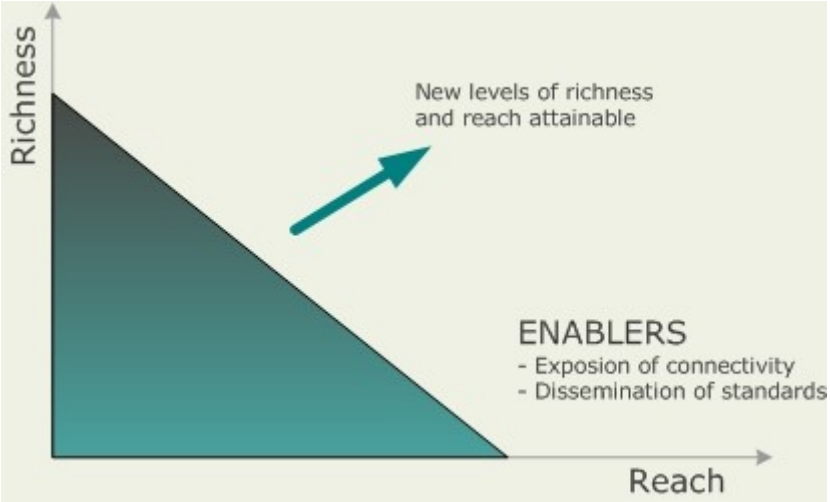


Figure 5. Information richness (Evans & Wurster, 2000)

The Open Systems Database Association (2006), Hewitt and de Jong (1983), Ohya (2005) and many with them point out that the communication in open systems needs to be on the basis of *distribution* and *dissemination*. Dissemination is described by Hewitt and de Jong (1983) as the electronic representation of publishing in a magazine or bulk mailed advertising. Once a piece of information and/or knowledge is disseminated and distributed, other knowledge in the system might be affected and changed. In open systems the distributed database approach does not work very well because of rights and privacy issues. For example when a proprietary software system is distributed, the organization distributing the software cannot in general access the copy on the customers' premises in order to make updates. Instead revisions and extensions must be disseminated for the customers to integrate into their own systems.

Service management is today a widespread concept and a big part of the management domain. Although service management is not clearly defined and there is no precise definition of what the theory of service management means or actually involves. One should consider it as a management perspective that should be viewed both from IT and business. Service

management is a total organizational approach that states the *utility of service*, as perceived by the user, and the *value of the service* to the organization. (Albrecht, 1988)

Just like service oriented architecture, service management has a number of fundamental objectives an organization should carry out to increase value. The objectives of service management can be applied to every organization that handles IT and IS (Gronross, 1986). Organizations need to understand the usefulness end users receive by consuming a service and how these services contribute to the value which the organization obtains. That is, management and participants must understand how usefulness is perceived in user relationships, and how relationships and behaviors changes over time.

Management and involved people have to understand how, personnel, technology and physical resources, and applications i.e. the IS organization, will be able to generate and distribute this usefulness and value. The last two objectives are associated with development and management of the IS organization. Develop and manage so that the planned effectiveness and value are achieved and design the operations of the IS organization with all parties involved and all of the requirements are identified. (Gronross, 1986)

2.10.1 Managing Application Services

One way to manage services is, according to Bianchini et al (2005) the concept of *applications services portfolio*. Applications services portfolio is a powerful instrument for understanding and managing information handling services. Application services can be categorized into strategic, high potential, key operational or support depending on their current or expected contribution to business success. Organizations can divide their services into specific categories depending of their usefulness. The framework of services portfolios suggests that services are categorized into four basic subject areas – strategic, key operational, support and high potential services.

- The first area is called the strategic area and helps organizations to manage and control their business and processes. The services in this category are core business processes that create or support change in how the organization conducts its business, with the aim of providing competitive advantage. These services are critical to future business success. Note that whether the technology used is 'leading edge' does not indicate that the application is strategic – assessment must be based on business contribution.

- Services and/or applications that maintain the existing business processes. *Key Operational services* are helping in the avoidance of any disadvantage.

- In the s*upport* application and services group, identified processes that are translated into IT services which improve business efficiency and management effectiveness are stored. They do not sustain the business or provide any competitive advantage.

- *High potential*: innovative information handling services which may create opportunities to gain future advantage, but are as yet unproven (Bianchini et al., 2005).

2.10.1.1 Service Quality

A central aspect of the management of all services is attending to their dimensions. However, it is important not to mistake *'quality of work'* for *'quality of service'*. Over the last 20 years, the marketing and quality literatures have devoted considerable attention to the issue of service quality and delivery. Within the quality movement there are the notions of the

'internal customer' and 'service level agreements' or 'service guarantee' which are often devised to a set parameters around the expected deliverables or outcomes

Quality is usually assessed based on conformance to expectations and requirements, i.e. does the service meet organizational, customer, employee, stakeholder and other involved expectations and requirements? According to Bianchini, 2006 there are two dimensions that determine whether or not the actual service meets the participant's expectations;

- *Technical quality*, is concerned with what the user receives and when. Technical quality is essentially about the specification of the service, defining what it is and what that the user can expect and receive – often called the service task (see section 2.10.1.2).

- *Emotional quality* relates to how the participant experiences the delivery of the service. This dimension is generally concerned with how customers want the service provider to treat them. Therefore, emotional quality is directly related to the process of service delivery and the extent of user involvement in that delivery process.

The challenge with many services is to recognize this emotional aspect, particularly in situations where there is a high degree of user involvement in the delivery of services. Value should not be confused with quality. Value is concerned with the overall benefit or satisfaction that the organization derives from its investment in the service. An increase in service quality, often incurs a cost were an improvement in the effectiveness of users may not result in an increase in value to the business (Bianchini et al., 2005).

2.10.1.2 Service Task

The service task states why a service exists and what the usefulness and value of the services are. It is a statement that conveys the essence of what the service delivers and thus provides both management and the workforce, both business and IS, with a goal to achieve. Note that the service task should always be defined and designed with reference to the user, the organization and its overall value to the business and to the business participants. To understand the service task of a specific service the organization needs some guidance and Bianchini, (2005) recommend three questions to concentrate on, both by management and technicians. The answers to these questions should be as concrete as possible. Key questions for service task definition:

- What are the important elements of the services to be provided, stated in terms of results, produced for the user, for IS staff and the organization as a whole?

- How are these elements supposed to be perceived by the target 'community of users'?

- What efforts does this suggest in terms of designing, delivering and marketing the service?

2.10.1.3 Service Standards and delivery system

Service oriented architectures are based on the service oriented methodology and standards such as XML are an essential part of the architecture for data representation and communication. Standards are measurable and typically articulated as service level agreements (SLA;s) The service delivery system specifies how the service is produced, including how it is *controlled* for quality, cost, and satisfaction. It essentially determines the

combination and configuration of people to be used in delivering the service and achieving the service standards. Erl (2005) say that the service delivery system must be synchronized with the service task and service standards so that the service encounter remains a pleasurable one for all concerned. The design of the service delivery system will determine the role that both the provider and recipient of the service can be expected to play in the delivery of the service.

Often, as it is not made explicit to users, they do not understand their role in the process and consequently this can impact the quality of service as perceived by them. (Bianchini et al., 2005)

2.10.1.4 The Dimensions of Services and the value of the service

Outside the *service task*, which provides a high-level description of what the user can expect of the service exists "*exploration"*. Exploration indicates that services can be 'unpacked' into a number of dimensions. This has been previously done for product quality (Garvan, 1984) and service quality (Sylvestro et al., 1992).

There are two aspects to consider in addressing the value dimension: the individual user or community of users availing themselves of the business as a whole. At an individual level, we must address the utility that the user is seeking, and recognize that this may or may not be directly related to their job. At an organization level, the benefit may be different. The cost of providing a service must therefore be matched against the utility and organization al benefits, with the overall value being a function of the utility and benefit. (Bianchini et al., 2005)

2.10.1.5 Handling of services and information

The form in which the service is required also has two aspects, first there is the nature that defines the expected service i.e. the service task and how this relates to its value. Second there are issues concerning *responsiveness e.g.* how flexible is the service? Is the service to be customizable or is it a case of 'one size fits all'? The form of the service is connected to the level of service and the level of service has three aspects recommended by Bianchini et al., (2005) and these three parameters are similar as for the concept of Information Handling. Information is created in many different places and one has to know that the information is accessible, available and reliable (SIG Security, 1997).

- *Access*: How easy is it for the users to gain access to the service?
- *Availability*: Is the service available when required?
- *Reliability*: Required reliability in terms of consistency. Is the service 'experience' similar every time it is delivered? (Bianchini et al., 2005; SIG Security, 1997)

Seen from a more software perspective there are some aspects to consider in the design of the delivery system of services;

- *Assurance*: confidence of users in IS staff and in their ability to deliver the service.
- *Empathy*: friendliness of IS staff and their ability to identify with the problems which the user is experiencing.
- *Involvement*: degree of expected user involvement in availing of the service. (Bianchini et al., 2005; SIG Security, 1997)

2.10.2 Chapter summary and highlights

Service management is not an individual area or a single theory of management; relatively, it is a management perspective. Although service management is not plainly defined and there is no exact description of what the theory of service management means or in fact involves. Identical to service oriented architecture, service management has a number of essential aims an organization should complete to raise value. Consider service management as a management perspective that ought to be viewed both from IT and business. The objectives of service management can be useful to each organization which handles IT and IS. Organizations need to understand the usefulness end users receive by consuming a service and how these services contribute to the value which the organization obtains. Service management intends to supply a decision-making standard from a process-related perspective, and to provide professional realization.

2.11 Product life cycle

A product's life cycle (PLC) is made up of the activities that go into making, using, transporting and disposing of a product. The life cycle is commonly shown as a series of stages, from "cradle" (raw material extraction and harvesting), through fabrication, manufacturing, packaging, transportation, consumption, and recycling, to the grave (National Office of Pollution Prevention et al., 1997). Most discussions of product life cycle portray the history of typical product as following an S-shaped curve. This curve is typically divided into four stages, known as *introduction*, *growth*, *maturing* and *decline*. Each performance form, organization, or specific offering, and all levels of the product may be subjected to the cycle analysis (Kotler et al., 1997).

One key concept is that a life cycle indicates that product life spans are limited; no product has an endless future. Informed intentional marketing can, however, maximize profit potential and time spent at each life stage. A second key concept is that the organization will need to develop new marketing strategies for product several times over its life span. For example, when a product is introduced, the marketing strategy should be consistent with eventual product positioning. As sales grow, the product may be improved, and new features may be added. When sales grow, flattens products have entered the mature stage. At this point, organizations may try to increase usage of the product or find new markets. Finally, sales decline and the firm must decide whether to invest in an attempt to renew the product, or divest in order to reduce losses. Periods of growth, maturing, and decline industry sales and profits represent different phases of the product life cycle. An economic sector’s historic sales and profit patterns can reveal the beginning of maturity or decline. Changes in the level of sales growth are often associated with changes in what it takes for a firm to be successful in an industry. (Rea & Kerzner, 1997)

PLC describes typical sales and profit patterns that occur over the life of a product or service. Product sales tend to pass through recognizable stages that require different manufacturing, financial, purchasing, personnel, and marketing strategies. As a marketing planning tool, PLC helps to outline potential marketing strategies. However, applying PLC concepts to predicting is not particularly useful. Critics claim that sales histories are too diverse, and that stages vary in length for different products. Thus, while it has its purpose, PLC concepts ought to be applied with care. (Rea & Kerzner, 1997)

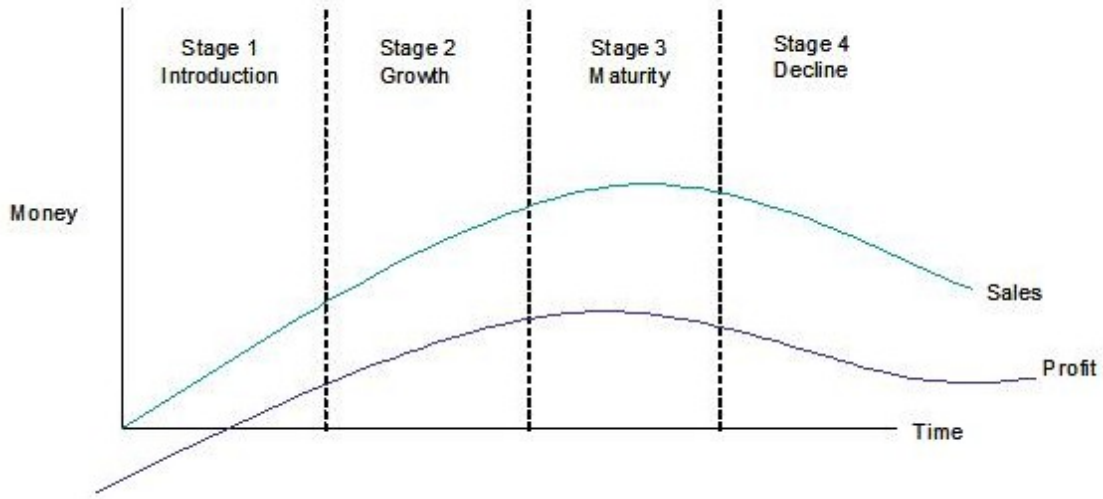


Figure 6. Product Life Cycle (Kotler, 2000)

2.11.1 Life cycle management

Generally, life cycle management starts early and ideally is integrated into the entire development cycle itself. That is, even before the product has been developed, the people responsible for development and marketing must have in mind the need to provide an adequate life to the product (Voet, 2005). Box (1983), Day (1981) and Levitt (1965) all say that the conditions a product is sold under will change over time. PLC refers to the succession of stages a product goes through. PLC management is the series of strategies used by management as a product goes through its life cycle.

Life Cycle Management (LCM) is a business decision-making approach that considers benefits, costs and risks over the full life cycle of a product or service. Life cycle management considers the product life cycle as a whole and optimizes the interaction of product and service design, manufacturing and life cycle activities. The goal of this approach is to protect resources and maximize effectiveness by means of life cycle assessment, product data management, technical support and, last but not least, life cycle costing. (Proceedings of the Institution of Mechanical Engineers, Part B (2001)

Government, business and non-governmental organizations can apply the life-cycle concept to their decision-making processes related to environment and product policy, design, and improvement. The life-cycle approach can also be used as a scientific tool for gathering quantitative data to inventory, weigh and rank the environmental burdens of products, processes and services. All organizations make decisions that can influence the inputs and outputs of upstream and downstream stages of the life cycle. Examples of such decisions include (National Office of Pollution Prevention et al., 1997):

- which product(s) to manufacture
- design of the product(s)
- types of feedstock to be used
- sources of supply of the feedstock
- sources of energy to be used
- type and amount of packaging
- management of manufacturing wastes

- instructions given to users
- management of post-use wastes.

Taking an LCM approach can change the decision-making process and result in altered outcomes. Life cycle management is simply about making this happen in a more planned and systematic way. Because it is applicable to a wide range of decisions, LCM is not a task to be assigned to one individual or office in an organization. It is a commitment, mindset and approach to be adopted throughout the business. (National Office of Pollution Prevention et al., 1997)

2.11.2 Service life cycle

Services in this thesis are in analogy with Cordeiro and de Carvalho (2002) all of the organizational activities and processes an enterprise can identify and further divide into sub processes such as business activities, business transactions and system activities enabled as IT services. Services work together in a network of software components that can receive a call and provide a given behavior on behalf of the calling system, application or component. One should especially note the difference between services.

A service is an abstract resource that represents a capability of performing tasks that form a logical functionality from the view of provider's entities and requester's entities. To be used, a service must be realized by a concrete provider agent. A service is usually available when needed, but remains idle until a request arrives. New services can be offered by combining existing services, such as the post office that is now using transportation services (Dubray, 2005). A service life cycle is expressed in the state transition diagrams below. There are two separate transition paths: *service* itself and *request processing.*

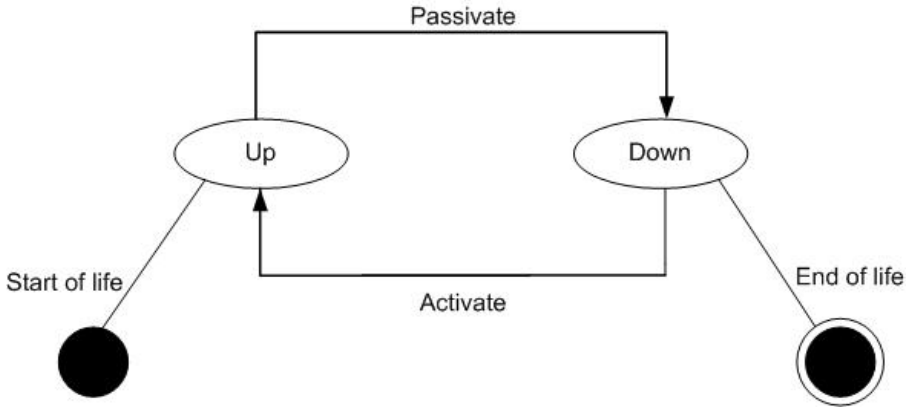


Figure 7. Service (W3C Working Group Note, 2004)

States:

- Up (compound) - the provider agent is capable of accepting and processing requests (the service is available).
- Down (compound) - the provider agent is not capable of accepting any requests (the service is not available).

Transitions:

- Start of Life - the service starts its life in Up state.
- End of Life - the service ends its life from Down state.
- Activate - the service can become available which transitions it from Down to Up state.
- Passivity - the service can become unavailable which transitions it from Up to Down state. (W3C Working Group Note, 2004)

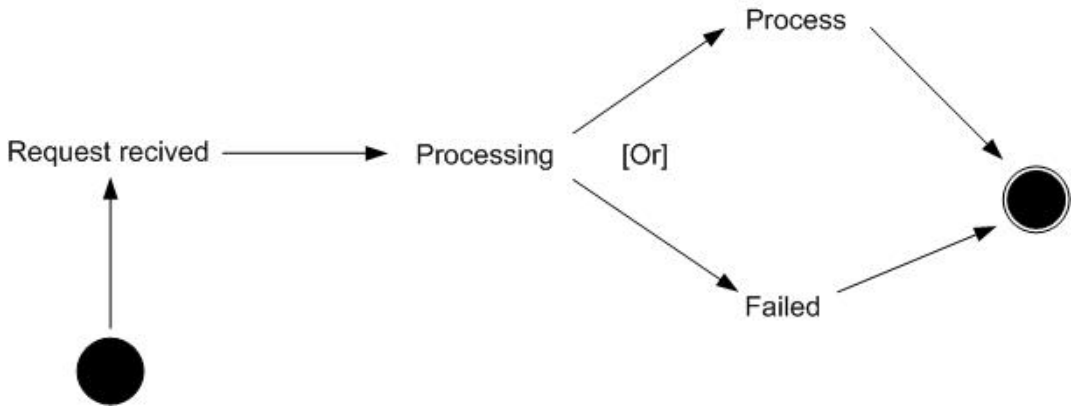


Figure 8. Request Processing (W3C Working Group Note, 2004)

States:

- Request Received - the provider agent has accepted a request to perform one of the service's functional responsibilities.
- Processing - the provider agent is doing some internal processing/execution to fulfill the requested function.
- Processed - the provider agent successfully completed requested function returning results to the requester agent.
- Failed - the provider agent encountered an error and didn't complete the requested function, returning error to the requester agent.

Transitions:

- A service starts request processing when it accepts (receives) a request.
- A service starts execution (actual processing) after it received a request.
- A service transitions to either processed or failed state depending on the outcome of the processing/execution stage.
- A service exits request processing from either processed or failed state (which are mutually exclusive according to the previous transition).

2.11.3 Chapter summary and highlights

Life Cycle Management is a business decision-making approach that considers benefits, costs and risks over the full life cycle of a product or service and it is in fact a concept and a research field of great relevance for example; SOA governance establishes strategy and processes, the decision rights and accountability for the whole service lifecycle including business analysis, service modeling, the SOA platform infrastructure and its tools, and also

the services realization and operations. The service lifecycle is highly noticeable because decision rights are clearly defined and followed, and the impact of change is reduced by extending existing processes and organizations to manage the service lifecycle. Taking an LCM approach may transform the decision-making process and the consequence is altered outcomes.

2.12 Quality and quality of Service

The definition of quality can vary, from object to object, from service to service, from person to person etc. Service is an essential and distinguishing attribute of something or someone. The concept quality owns a lot of attributes depending on situation and behavior; quality can be *good, bad, positive, negative, superior, inferior etc.* (Word Net, 2006)

Winder (1993) adds; quality, in its maximum condition, exists in five divergent dimensions;

- With the knowledge from earlier problems, implementations, sales etc. one will get e*xperience*. Experience is acquired practical knowledge and in the experience dimension things are actually implemented and carried out. The experience will work as a blueprint or a roadmap for both people and organizations.
- The next stage is *measurement*s; were one measure how well or how poorly the task was implemented and its impact based on specific selection of parameters.
- By controlling and observing associations between objects e.g. people-to-people or system-to-system and people-to-system *relationships,* organizations, scientists, designer's etc can improve the *systems thinking.* System thinking and relationship analysis permits organization s to standardize systems so that continuous learning is retained.
- *Inter-connectivity and paradigm logic,* analysis of valid internal and external information through a new paradigm provides a new way of experience information. Through this new viewpoint new logic is earned, and this new logic provides a wholly new way of understanding and interpreting information. This new way of seeing information can be an advantage in the competition against other companies on the same market, e.g. to gain shares of the market.
- The last parameter can be described with the quotation; "*If I give you something that has more value to you than is does to me, then together we are better off as a result of the trade.*" (Winder, 1993, pp.1) The above quotation pronounces the value of sharing among people and among people and organization s. From an organization al point of view it is all about "the delight of customers". It is when considering the value of an exchange organization can produce consistency and improve products, services and better satisfy the customer's needs.

The issue of the five dimensions and quality parameters is, roughly expressed "delight the customer". However, the dimensions can not exist separately; they are all connected to each other in one way or another. The value sharing dimension is the blueprint for integration of all the other dimensions. Since these dimensions provide the structure for the service quality, a basic understanding of them is essential in fulfilling service quality. (Winder, 1993)

About earlier mentioned functional and non-functional (see section 2.8.3.4) and the recently described quality aspects, each service are characterized by a set of quality parameters. In service oriented architectures quality must be defined in both provider perspective and requestor perspective. The provider perspective defines the quality offered by the service. It is worth noting that the involved quality dimensions strictly depend on the application and

domain in which the service, service provider and channel operates. Since it is almost impossible to define a set of quality dimensions valid for all the possible services, according to each kind of service, a community exists that defines which the relevant aspects are in different situations (Marchetti et al., 2004). Task of this community is to *identify* the set of relevant quality dimensions for such particular class or services. Whoever wants to implement a service belonging to this class must refer to this set of quality dimensions in order to characterize the right quality and thus make discovery of services easier. In this way the service model includes on the provider side:

- QoS *(Quality of Service)* definition, in which a set of admissible values is assigned to the quality dimensions identified by the related community;

- QoP *(Quality of Provider)* definition, in which a set of admissible values is assigned to the quality dimensions defined by the provider community;

- QoC *(Quality of Channel)* definition, for each channel available to invoke the service, where a set of admissible values is assigned to the quality dimensions defined by the community of that particular channel.

About quality definition, whereas in the provider perspective the model defines the *quality offered*, here the service model identifies which are the user needs. The user must be aware about the existence of the communities and, in particular, about the *service* community. According to that, we assume that the user is able to define the most relevant QoS for each requested service. It is important to highlight that the service is connected to the user through different channels, which also influence the QoS.

According to van Moorsel (2001) Quality of Experience (QoE) can be described as a function *f(QoS, QoC, QoP)*. On the basis of the obtained values, the user will be able to define the admissible value of the quality of service dimensions for the service offered by a particular provider through a given channel. With the help of Service Ontology we can organize services into three different layers of abstraction to increase service discovery on the basis of user functional requirements in service oriented architectures. Further refinements of the discovery process can be performed by taking into account context information and, finally, selecting only those services that match user quality requirements. In the service ontology, we introduce three distinct concepts, according to an increasing level of abstraction (Bainchini et al., 2006).

The dimensions of service quality can inform the provider perspective in the following ways (Bianchini et al., 2005):

- *Value*: From a provider perspective, it may be that different user groups derive a different utility from the service. Therefore, this suggests that the IS function may need to examine the possibility of segmenting its user base to reflect this differing utility.

- *Form*: Depending on how form is defined by the user, the provider can then consider issues of service design and metrics.

- *Level*: The level of service guides pricing policies, access and availability policies and determines capacity requirements.

Delivery system: The definition of the delivery system in terms of assurance, empathy and expected involvement of users provides guidance for process design, HR policies, and the culture that needs to be looked after.

2.12.1 Chapter summary and highlights

The quality of service is essential in SOA; service is a necessary and distinguishing element of something or someone. The concept quality owns a lot of attributes depending on situation and behavior; quality can be good, bad, positive, negative, superior, inferior etc. But the description of quality can differ, from object to object, from service to service, from person to person etc. The quality must be defined in both provider perspective and requestor perspective.

2.13 Registry

2.13.1 SOA promotes service discovery

Though UDDI existed in the beginning of first generation of Web services standards but a small number of the first implementations used registries as part of the environment. It is possible that this is related to that not enough Web services were analyzed and designed to warrant a registry. Another reason is that the term service discovery was simply not designed into the architecture from the beginning. When operating within traditional distributed architectures, Web services were more often developed to supply point-to-point solutions and the complexity was not that huge. This is why service discovery was not a common concern. Today the problem is that the UDDI standard is not sufficiently extensive to support all of the parameters today's organizations have on their services. Management and governance includes a variety of artifacts needed to implement SOA. Now organizations have noticed that all of these artifacts need to be stored in a registry to gain control and surveillance of the enterprise. (Erl, 2005)

2.13.1.1 UDDI and rise of Service Registry

One present well used standard to maintain and preserve services is the so called UDDI[11]. UDDI is developed together with OASIS[12] and is an attempt to create a standard based on rules and regulations for easier service identification in information environments. It is important that such a register is based upon agreed standards because in the future more and more services will be developed and sold by more and more vendors. It there is no standard between these vendors and developers there will be difficulties in the integration and finding among services. One problem with UDDI is that it is not adequate and is not able to handle all of the requirements that the organization, stakeholders, users, partners etc. has on present and future services. (Abrahams, 2005)

Windley (2006) proclaims in his article Governance SOA:

> Registries are the primary tool organizations use for managing and communicating governance artifacts, as well as automating key governance activities. A registry provides a central reference or "system of record" for services. Think of it as a place where services can be advertised by providers and discovered by consumers inside an organization – a control point for governing service availability,

[11] Universal Description, Discovery and Integration
[12] Organisation for the Advancement of Structured Information Standards

> versioning, and compliance with internal and external requirements. (Windley, January 19, 2006, pp. 3)

One of the most important factors in a Service Oriented Architecture and the characteristics it provides is to handle and maintain existing services within the organization. The company must know which their services are to avoid redundancies in services. The organization does not want to develop a service that already exists. It might happen that organizations are not aware of the importance of tracking and controlling services in the beginning of a SOA implementation, when services exist in small numbers. As times go by and the organization grows and the demands of new services arise, suddenly there are more services than the organization expected and no one has a clue where they are or should be stored and managed for efficiency. Now the organization has moved the complexity to another level and reached service spaghetti. One way to create overview concerning services and their data/information is a service registry, where the organization centrally stores their services. (http://www.sun.com/products/soa/registry/)

Steve Edens, Enterprise Architect at Blue Cross Blue Shield, Massachusetts says:

> Service Registry is a critical component to an open source Governance strategy as it unifies the entire SOA framework. With this tool we are able to address all SOA domains including security, information management and orchestration, which in turn improve visibility and quality of information across our organization while mitigating the risks." Why a Service Registry and the value of a registry for SOA. (Edens, Sun's Service Registry, June 8, 2006)

A Services Registry is required when an organization needs to track and manage increasing numbers of services. The promise of services depends on the ability to share assets, which requires centralized facilities for access and control. Most typical Web Service registries today provide basic publish and discovery of service descriptions. Registries are used to store SOA artifacts (WSDL, Schema etc.) which are composed at run time to support BPMS[13] activity. (Hodge, 2005)

When using Service Registry it is appropriate to investigate who controls SOA metadata. Who is responsible for the service and the service life cycle? Who are the developers and the IT operations people? Which are the people in line of business? SOA committee? Etc. An organization must clear out who is liable according to policies to do what with a service to asses the proposed benefits of a service registry (software reuse, service catalogue, business agility, compliance). With the use of Service Registry, customers can truly address both Services access and SOA governance. (Effective SOA deployment using an SOA Registry Repository, 2005)

Service description is non trivial, and dependent on role, use case, and what value the constituency is trying to derive from the service. Many of the descriptions of a service require various types of policy enforcement mechanisms, and also some structure for metadata registry system. (Sim et al., 2005)

When the organization applies the principles of service orientation and begins to manage IT services as a portfolio, they receive a common language for communication and employees

[13] Business Process Management System

will much easier understand and identify services. Information-handling services are provided via the overall portfolio and should be built into the technology platform and organizations must not forget the importance of all their services as a whole to function efficiently. In present days there has been an increasing trend for provision of services from external third-party vendors and external service providers. Over the years the role of IT in business has expanded significantly to the extent that it now often shapes business strategy. Information is also recognized as a key corporate asset. (Erl, 2005)

2.13.1.2 Service Categories

In his article Managing IT as a Portfolio of Services, Peppard (2003) says:

> Research and practice form service industries highlights the inappropriateness of treating service as a homogeneous concept – different services can exhibit different attributes, raise different issues and require managing in different ways. Unfortunately, researchers have found that many organizations treat all services in a similar fashion and manage them in similar ways. (Peppard, 2003, pp. 470)

Organizations services can be labeled in four different titles according to Peppard (2003): *applications services, operational services, value-enabling services and infrastructure services*. Erl (2005) has a different point of view; he suggests that organization divides their services into three similar groups; application services, business services and orchestration/process services. This is only a semantic question and we assume that the meaning and purpose is the same, to create awareness of existing services and with the procedure of Separation of Concerns the organization breaks down the complexity of several services into smaller catalogues for easier management. In this thesis we will describe the services from Erls' point of view, with the supplemented information Peppard (2003) advocate.

Unlike application services, which are technical, are *business services* only concerned with business logic and the representation of the same. Business services are one of the most important objects of SOA (Erl, 2005). Service-orientation is the basic method for bringing the business model into Web services arena and it is the business services responsibility to express the business logic to other dependent services on lover levels. In non-technical terms are business services always an implementation of the business service model.

Orchestration/process services are often described as more valuable than a standard business process, because they allow process and business logic to be linked to service interaction. Orchestration services consist of one or more processes that compose business and application services according to business rules and business logic embedded within process definition (Erl, 2005). And Savarese (2003) add, orchestration services are positioned inside the so called orchestration layer, and their primary task is to connect services to other services.

Application services work in the so called application service layer and these services described as technical implementations of specific functionality and are often distributed via software applications. Erl (2005) speak of them as providers of reusable function related to processing data within new and legacy application environments. These services are derived from the 'information handling' abilities of technology, and include information processing services, information sharing services, information storage services and information access services.

In addition to Erl's definition of applications services Peppard (2003) has some information to add:

> Application services directly impact the performance of business processes with process designs or components of processes embedded in software applications. While organizations have traditionally custom developed or purchased 'packaged' application software, today, application service providers deliver a variety of information handling services direct to the desktop or other device via both fixed and wireless Internet for a rental fee. (Peppard, 2003 pp.470)

2.13.1.3 Service value and assessment

When organizations invest money in services they do this to best fulfill the long-term goals and the short-term goals. The problem in this service investment is that organizations do not treat services as every other investment; this can be due to the complexity of services and difficulties in characterizing them. It can also be that services often and in this case, in the environment of SOA, services are abstractions of business processes and therefore hard to describe and to see what cost and benefit they generate to the overall business objectives. One attempt to track the value of a service is to see what worth the consumer of the service obtain and how this worth is connected to the organization and their long and short-term goals.

Organizations must in order to be effective analyze which services they have and what they provide. The term usefulness from the financial world is one way to investigate the value customers receive when consuming a specific product and/or service and the pleasure they get from the consumption. (Garvan, 1984)

Usefulness can be increased through many different points of attack, and Woodruff (1997) suggests that decreasing of problems, avoidance of non benefits and increase in quality can improve consumer's usefulness. Usefulness often include a more 'soft' visual angle; the emotional aspect of customers. This emotional aspect involves the goals customers have consuming the service. Organizations must therefore, not only investigate what value the consumers receive but also their goals and objectives to get a better fundament in value description of their services and this will lead to better strategy when classifying the services.

2.13.2 Service level agreement

A service level agreement (SLA) is a document that provides a description of purpose and issue offered by existing and future services. It also includes desired service performance, guidelines to track services in the organizations IS/IT environment and rules concerning communication and information between different departments e.g. between business unit and the IT division. The latter is achieved through negotiations and leads to definition of provided services from IT and related service costs. In the discussions between involved divisions clarifies requirements, issues, objectives desired quality, expenses, foes, customer duties and responsibilities etc. and the result of this negotiations informs the organizations governance committee about decisions concerning infrastructure, architecture, and application needs. It is essential to use SLAs as a key ingredient when involving external partners such as providers. There are many possible outcomes of a partnership, if the result of the partnership does not result in decreased costs concerning providing internal services the organization must decide weather or not to outsource infrastructure services. The outcome should be as god as possible to the involved organizations. Business units can with the assistance of SLAs more carefully also request services from IT and force IT units to think like external providers. Internal IT divisions should "sell" their services to the business unit in the same manner they sell to external customers and thus should frequently look for ways to save money. As stated before,

it is essential that divisions speak the same language and the challenge in the SLA process is in translating business requirements concerning services into IT services in a way IT can understand.

Common costs related to the IT division come from work and from processing time, storage capacity. When business units need services such as processing of invoices, access to the web, and rapid response to queries, IT units must translate their costs into terms managers can understand. If the SLA only lists IT costs in terms that only IT people understand business units will make bad choices concerning IT service levels and use of services. But it will work the other way around, the different language for communication will not help IT service people and committees design shared services in the best way to succeed. SLA has succeeded when communications about business needs and IT services smooth the progress of decisions that lead to decreased costs and increased usefulness of both business and IT resources. As a result, well designed and well documented SLAs can support IT and business makes better choices – how to buy, sell, and price. This will lead to better IT services and better understanding among people working in both business and IT what value generated by developed services. (Ross & Weill, 2004)

2.13.3 Chapter summary and highlights

A Services Registry is obligatory when an organization needs to track and manage growing numbers of services. Organizations must not underestimate the importance of functioning services. Most Web Service registries today provide basic publish and discovery of service descriptions. By taking on service orientation and administrate IT as a portfolio of services, it grants a language and perspective that all employees can identify with and comprehend. Information-handling services are delivered through the organizations portfolio of applications that are implemented on its technology platform. Today a Service Registry is much more extensive than UDDI and provides services that are needed to successfully manage complex SOA implementations.

Service level agreements list available services, alternative quality levels, and related costs. Through consultations between the IT service unit and business units, an SLA show the way to expression of what the services IT offer and the costs of the services. SLAs often support comparison with external providers. The challenge of the SLA process is the translations and communication between business service-level and IT.

2.13.4 Chapter closure - theoretical frame of references

To summarize the theoretical frame of references chapter we present our theoretical model again to facilitate the comprehension and tie together the theoretical material we presented in this chapter, hopefully the model emerges extra clearly and you understand our viewpoint and declarations of the fundamental parts of this thesis.

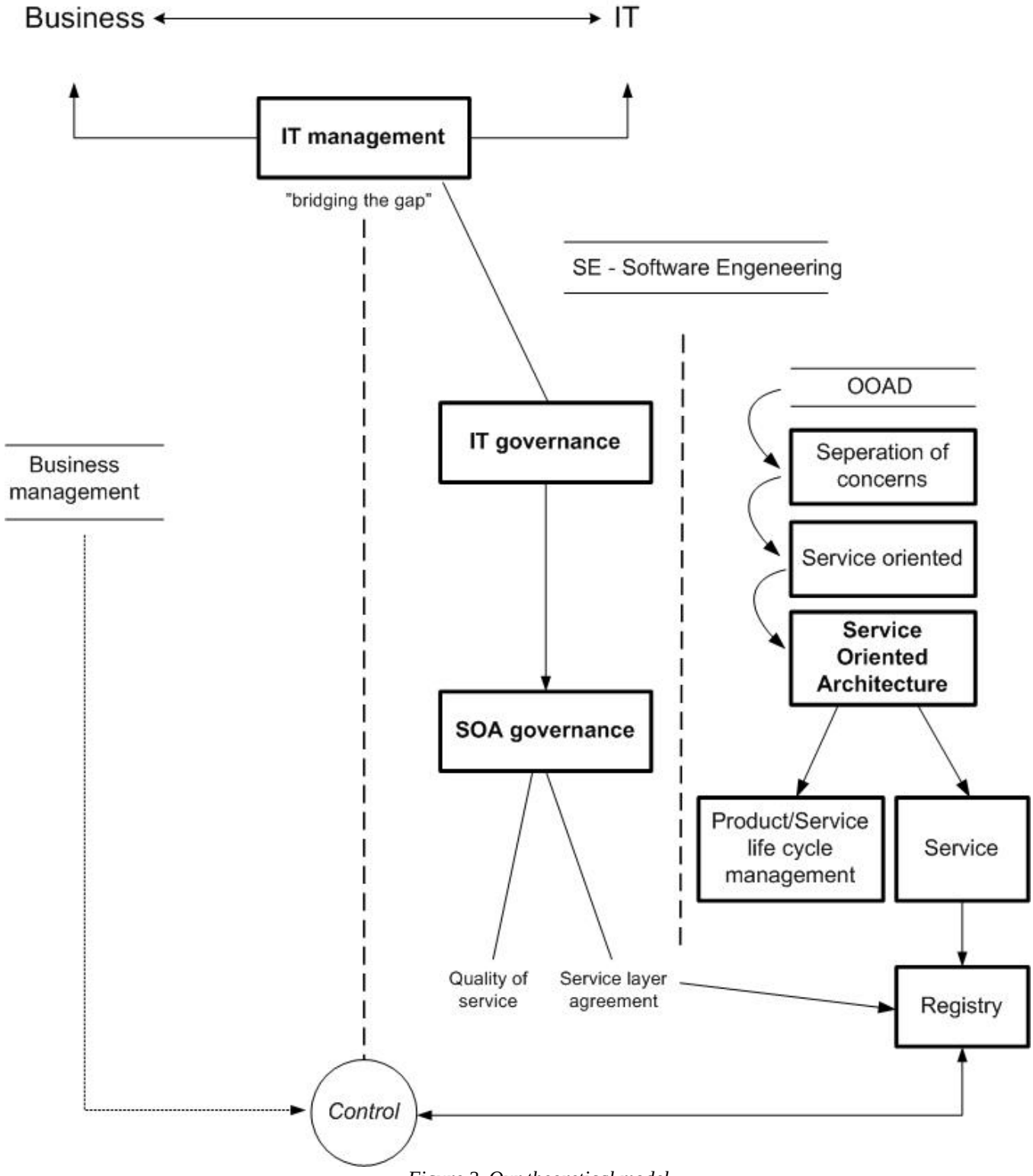


Figure 2. Our theoretical model

3 SCIENTIFIC APPROACH

This chapter presents our line of action and statements to why we selected a certain approach. This thesis was performed through a number of stages. Based on our labor and accomplishment we designed this model were the pieces of the thesis are connected and related to each other in one way or another. For example, the *interviews* are based on the *scientific approach,* the *purpose and the issue* is a foundation for the *introduction and background* chapter.

We had to choose, prioritize, and terminate decisions for best progress to the centre of the model. Literature has been reviewed with the hypothesis in mind the result of the theoretical frame of reference has been discussed and analyzed together with the case study material from the interviews.

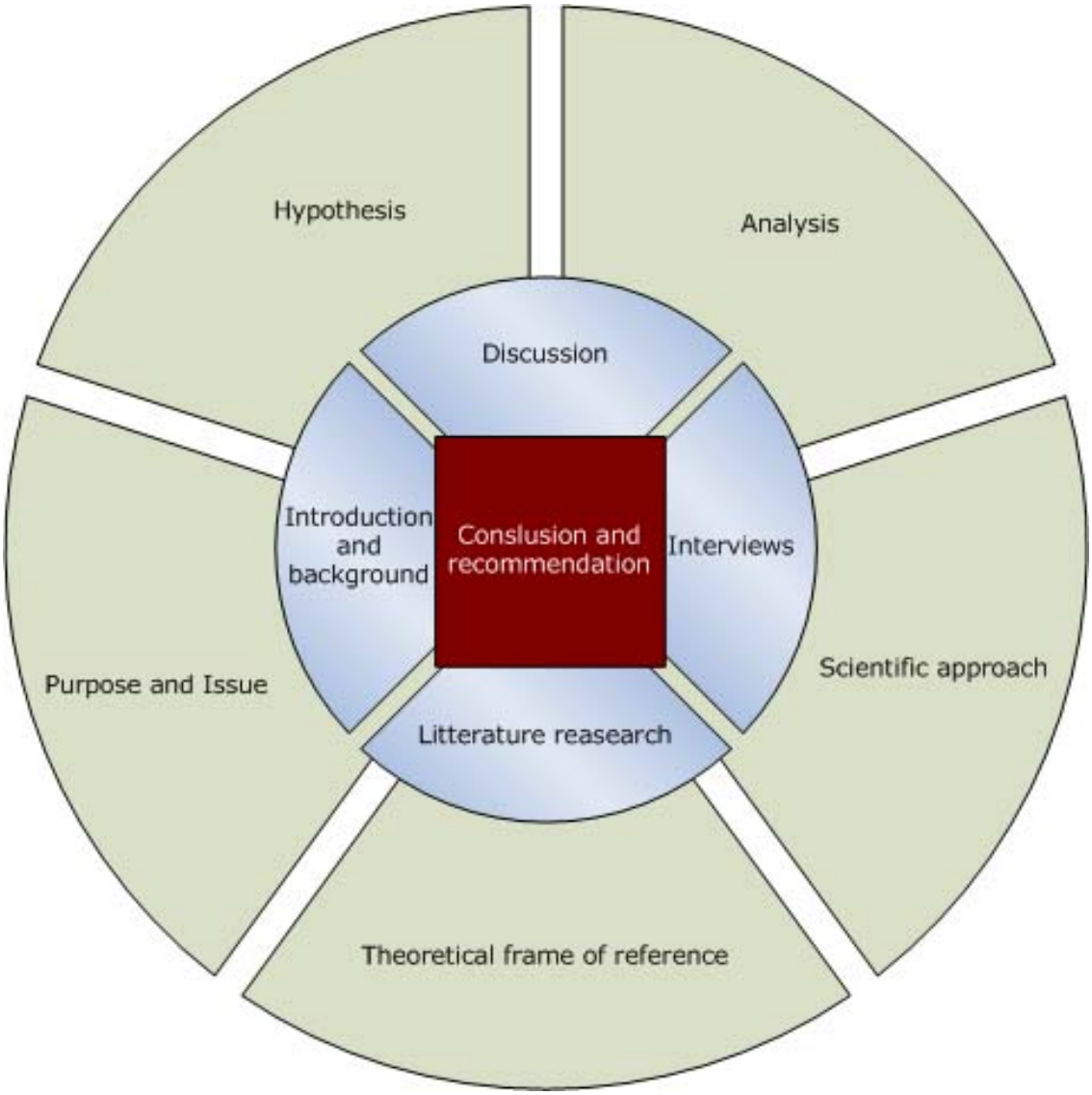
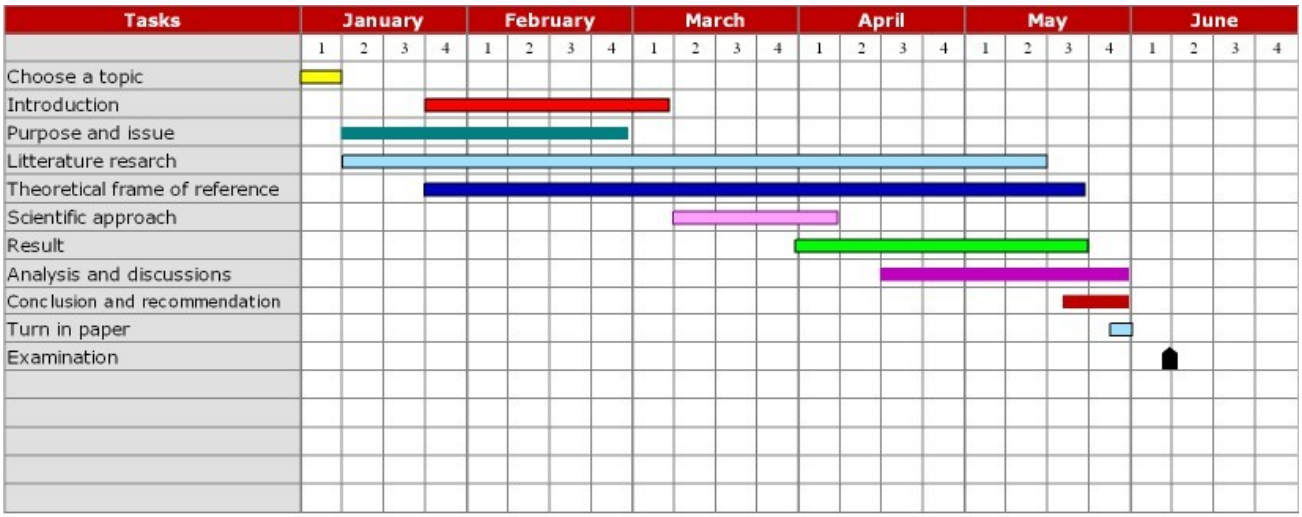


Figure 9. Master thesis line of action

3.1 Scientific framework

The picture below illustrates our time plan and point up how we divided each process in this project and planned our time thereafter. Key dates were selected to support the projects objectives. We reached the time plan to approximately 90 percent; the main problems were mostly related to our massive research of literature used for the theoretical frame of references which demanded a huge amount of time. Moreover we achieved seven interviews, our goal was to perform ten to fifteen, but despite of that we are satisfied with the outcome, we realized that ten to fifteen interviews are slightly over ambitious.



Key Dates	
19/1	Initial meeting with IBM
1/3	First project check up
27/4	Second project check up
5/4	Start of interviews
19/5	Turn in first copy of thesis
31/5	Turn in paper

Figure 10. Project time plan and course of action

3.1.1 Qualitative and quantitative

The scientific approach consists of two basic lines of actions; qualitative and quantitative methods. Qualitative methods concerns, according to Easterby-Smith, Thorpe, and Lowe, (2002) in-depth interviewing while quantitative methods concerns surveys and interviews on a more superficial level. Instead of in-depth interviews, surveys are used to cover up a wide spectrum of results. Qualitative research is referred to in the literature by a number of terms, including naturalistic inquiry, ethnographic research, field research or field work, unobtrusive measures, observation, grounded theory research, and interpretive procedures (Glazier & Powell, 1992)**.**

Other authors, Holme and Solvang (1997), describe qualitative methods as finding characteristics of the respondents and the life situation of their organization. The importance of information in qualitative methods is very dependent on the information source i.e. the respondent. Information is often gathered during interviews in a way that are very alike regular and common conversations. In this study, a qualitative approach was used founded on the numerous arguments presented in this chapter. Moreover, we want a qualitative approach

in view of the fact that the closeness with individuals and integration of people's thoughts based on their experiences. We prefer to study unique individuals were every person is selected because of their knowledge of the chosen phenomena or subject matter. We also promote *holism*[14]- to see the overall picture. In qualitative method the context is of importance, another motive to why we prefer a qualitative method. With perspective awareness we present our understanding, we believe that we applied the right analysis method and have good quality on the data, since we support every conclusion on quotation and references.

3.1.1.1 Qualitative Methods

Qualitative research is establishing itself in the social sciences and in psychology. There is now available a great variety of specific methods each of which starts from different premises and pursues different aims. Each method is based on specific understanding of its object. The most important procedures for collecting and interpreting data and for assessing and presenting results will be located in this framework. This will provide readers with an overview of the field of qualitative research, of concrete methodological alternatives and of their claims, applications and limits (Glazier & Powell, 1992). This makes it possible for us to choose the most appropriate methodological strategy with respect to out research question and issues.

Perhaps the best way to understand what is meant by qualitative research is to determine what it is not. It is not procedures that predominantly rely on statistical analysis for inferences. It is not at set of procedures that rely predominantly on quantitative measures as a means of data gathering. It is not a set of preliminary data-gathering procedures intended to be used as a device for determining what non qualitative methods should be employed for a project (Glazier & Powell, 1992). Qualitative research can be understood in terms of the means of collection and the type of data collected. These include observations, interviews, and content analysis of documents, articles, books, audio- and videotapes, even descriptive statistics such as census data. It includes not only "research about persons' lives, stories, behaviors, but also about organizational functioning, social movements, or relationships (Strauss & Corbin, 1990).

Qualitative data can include interactions between individuals, groups, and organizations as well as descriptions of phenomena. The strength of qualitative data is its rich description. Researchers struggle to capture the essence of a subject by using description that yields generalizations documented by specific examples of data from the field. Qualitative data can be said to yield more contextually detailed data – *richer* data. The richness of the data is ensured by the breadth of the context captured with the data. What this means is that, along with the data collected, one also gets a description of what activities were going on around the phenomenon in terms of time and social circumstances surrounding it. Researchers must be aware of and have access to any array of research tools that include the research methodologies and the means of analysis that are at their disposal. (Glazier & Powell, 1992)

[14] Holism (derives from the Greek *holos*, "whole", "undivided") is in common sense a philosophical way of looking at things that the sum of individuals are more than the parts and that nothing can be described distanced from its context (Lübcke et al., 1988).

3.1.1.2 The relevance of qualitative research

As the complexity of society increases, the need to develop efficient management and distribution of information resources becomes more and more critical. It can be argued that information is the substance on Adam Smith's metaphorical "invisible hand", which explains that you are guided by an 'invisible hand' toward ends that might not originally been intended, is grounded (Skinner, 1970). It is information, both as an abstract concept and a reusable resource, which has been the foundation of the development of modern society. Because of the complex nature and variety of roles that today's information professionals play in the research, it is critical that they are aware of and have access to the variety of research methodologies and means of analysis that researchers in other disciplines are employing. The use of qualitative research methods is a case in point. Qualitative methods are becoming more accepted among researchers in many disciplines, particularly in the social sciences. Complexity of society cannot always be understood from a single perspective therefore multiple perspectives have become an important element in modern social science research (Glazier & Powell, 1992).

Qualitative research is of specific relevance to the study of social relations, owing to the fact of the pluralization of life worlds. Key expressions for this pluralization are the 'new obscurity' (Habermas, 1996***)***, the growing 'individualization of ways of living and biographical patterns' (Beck, 1992***)***, and the dissolution of 'old' social inequalities into the new diversity of milieus, subcultures, lifestyles and ways of living (Hradil, 1992). This pluralization requires a new sensitivity to the empirical study of issues. Advocates of postmodernism have argued that the era of big narratives and theories is over: locally, temporally and situational limited narratives are now required, with regard to the pluralization of lifestyles and patterns of interpretation in modern and post modern society (Flick, 2002).

The flexibility afforded by having multiple approaches from which researchers may draw conclusions gives researchers a better opportunity to understand the complex social phenomena with which they are faced. (Glazier & Powell, 1992)

3.1.2 The Changing environment of information management

Rapid social change and the resulting diversification of life worlds are increasingly confronting social researchers with new social contexts and perspectives. These are so new for them that their traditional *deductive* methodologies – deriving research questions and hypotheses from theoretical models and testing them against empirical evidence – are failing in the differentiation of objects. Thus, research is increasingly forced to make use of *inductive* strategies: instead of starting from theories and testing them, 'sensitizing concepts' are required for approaching social contexts to be studied. (Geertz, 1983)

Higher education, corporations, governmental agencies, and even local school systems are increasingly concerned with the role and impact of information as agent for both local and global change (in the 1990s). Information resource management (IRM), a concept that emerged in the 1980s, continues to evolve and represents a convergence of perspectives. The integration of content, technology, people, and processes: the existence of at high-level executive, generically referred to as the chief information officer (CIO) (Barone, 1989) and the recognition of information as the fourth global institutional resource are considered IRM's cornerstones (Owen, 1989). Its success in the future, some believe, however, hinges on its ability to also incorporate end users in the total information management framework (Trauth, 1989).

3.1.3 Interviewing

As Benney and Huges (1970) point out, the interviewing is the "favored digging tool" of social researchers. When most people hear the term interviewing, they think of structured research tools such as attitude surveys, opinion polls, and questionnaires. These interviews are typically administered to a large group of respondents or subjects. People may be asked to rate their feelings along a scale, select the most appropriate answer from among forced-choice responses, or respond to a predetermined set of open-ended questions in their own words. (Benney & Hughes, 1970)

Although these research approaches differ in many respects, they all adopt a standardized format: the researcher has the questions and the research subject has the answers. In fact, in most structured interviewing each person I supposed to be asked identically worded questions to assure comparable findings. The interviewer serves as a cheerful data collector; the role involves getting people to relax enough to answer the predefined series of questions completely. (Taylor & Bogdan, 1998)

In contrast to structured interviewing, qualitative interviewing is flexible and dynamic. Qualitative interviewing has been referred to as nondirective, unstructured, non standardized and open-ended interviewing. The term *in-depth interviewing* is often used for the qualitative interview. By in-depth qualitative interviewing, we mean repeated face-to-face encounters between the researcher and informants directed toward understanding informants' perspectives on their lives, experiences, or situations as expressed in their own words. The in-depth interview is modeled after a conversation between equals rather than a formal question-and-answer exchange. (Taylor & Bogdan, 1998)

3.1.3.1 Types of interview studies

Three closely related types of qualitative interview *studies* can be distinguished;

- The Life history or sociological autobiography
- Interviewing toward learning about events and activities that cannot be observed directly, Yield a picture of a range of settings, situations, or people.

In *the Life history or sociological autobiography* – the researcher attempts to capture the salient experiences in a person's life and that person's definitions of those experiences. The life history presents people's views on their lives in their own words. (Shaw, 1966)

- Interviewing toward learning about events and activities that cannot be observed directly – in this type of interviewing, the people being interviewed are informants in the truest sense of the world. They act as observers – eyes and ears in the field – for the views, but to describe what happened and how others viewed it.
- Yield a picture of a range of settings, situations, or people – Interviewing is used to study a relatively large number of people in a relatively short period of time compared to what would be required in participant observation research. For instance, several in-depth interviews with twenty teachers could probably be conducted in the same amount of time in would take to conduct a participant observation study of a single classroom.

Although researchers select in-depth interviewing for different purposes, the basic interviewing techniques are similar for these different types of studies. In each case, interviewers try to establish rapport with informants through repeated contacts over time and

to develop a detailed understanding of their experiences and perspectives. (Taylor & Bogdan, 1998)

3.1.3.2 Choosing to interview

Thus no method is equally is suited for all purposes. The choice of research method should be determined by the research interests, the circumstances of the setting or people to be studied, and practical constraints faced by the researcher. In-depth interviewing seems especially well suited in the following situations.

- The research interests are relatively clear and well defined – Although research interests are necessarily broad and open-ended in qualitative research, the clarity and specificity of researchers' interests vary. Interviewing is well suited for studies in which researchers have a relatively clear sense of their interests and the kinds of questions they wish to pursue.
- Settings or people are not otherwise accessible – in-depth interviewing is called for when a researcher wishes to study past events or cannot access to a particular type of setting or people.
- The researcher has time constraints – it takes time to locate settings, negotiate access, arrange visits, and get to know informants. Although interviewers can face similar problems, studies based on interviewing usually can be completed in a shorter period of time than those based on participant observation. Interviewing makes the most efficient use of the researcher's limited time.

The researcher is interested in understanding a broad range of settings or people – In qualitative research, an "N of 1" can be just as illuminating as a large sample. However, there are instances in which the researcher may want to sacrifice the depth of understanding that comes with focusing intensively on a single setting or person for the breadth that comes with studying a range of places and people. (Taylor & Bogdan, 1998)

Interviewing multiple informants lends itself to building general theories about the nature of social phenomena. Analytic induction is one method of constructing theories from qualitative data that requires a sizable number of cases. (Robinson, 1951; Turner, 1953)

3.1.3.3 The interviews course of action

The interviewing informants for this master thesis were contacted by mail as a first action; the selection was organized by our supervisors at IBM since they have great understanding in relation to who possess relevant competence. To confirm interest, date and time we pursued with phone calls. The respondents were selected by professional position and the level of knowledge about service oriented architecture, for the reason that we want to widen the knowledge of SOA in different levels within the organization and to observe if the opinions differ from each level. To reach a result and to limit travels as well as accomplishing more interviews we finally choose respondents positioned in the area of Gothenburg. The interviews were individual and took place at IBM. A tape-recorder was used to support the information and the translation from Swedish to English.

We completed totally seven interviews; the length of the interviews was approximately one hour. After every interview we discussed the answers and the general impression. In section '

Appendix' the interview questions and a declaration to why we asked the questions are enclosed (8.1) besides that an example of an interview informant form cover, which were used to facilitate and organize our work (8.2). As a preparation to the analysis and discussion chapter we categorized the respondent's professional position and the questions in tree respectively five categories. We did this to summarize and analyze the respondent's understanding about service oriented architecture in relation to different levels; line of business, architectural and technical.

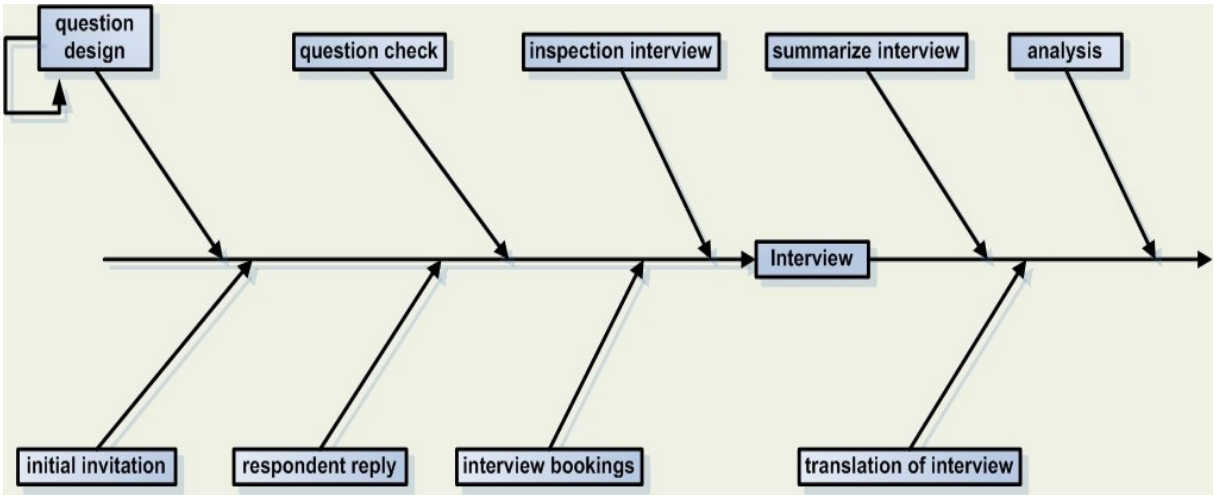


Figure 11. Interviews line of action

4 CASE STUDY RESULT

This chapter presents the gathered information from our interviews at IBM which are based on the theoretical frame of references. The respondent's answers are in this chapter presented under each question, we refer to the respondents by respondent A, respondent B, respondent C etc. Declaration of the respondent's professional position:

Respondent A - Consultant generalist focus on business values
Respondent B - Business and Enterprise Consultant
Respondent C - Account manager
Respondent D - Chief IT-architect, Operational aspects
Respondent E - Senior IT-specialist
Respondent F - IT-specialist, Tech
Respondent G - Technical chief

4.1.1 What is your understanding of SOA?

4.1.1.1 Respondent A

SOA is a concept which aims to create flexibility and a cost effective way to integrate different applications which will make it easier to maintain and connect new and old applications, and to standardize the management of the information architecture.

4.1.1.2 Respondent B

I see SOA out of a business perspective. SOA is all about business services; it is not technical IT solutions that build up services and creates the infrastructure of an organization. One way to understand SOA is to perceive it as the top of a pyramid. From the top things will generate, and you can study it as different layers were business services shall deliver value on the way to the top of the pyramid were business services are born and live. When you have your business services, you can break down the business service in a couple of sub services. After this you are able to implement a middleware containing a couple of services for integration. The next layer is the application layer and last the data ware.

The organization should along the way be able to look back and ask themselves way a specific service evolves and there should always be a business purpose. A service must fulfill a number of business requirements and there should be a number of parameters for service discovery.

4.1.1.3 Respondent C

SOA is an architecture where forces on the market are met. To meet the changes on the market one must have an underlying system that supports changes in the IT system. SOA is an architecture to realize those rapid changes.

I don't directly work with SOA anymore, I work with business driven development, this means that I work with applications development and systems development as an account manager within Rational on the Web Sphere area. If we think of the time when IBM began their definition of this, they talked about business on demand, and they still do. Independent of place, time, and terminal I should reach the IT systems and receive the information I need to complete my tasks. This was the first step towards SOA. Next step was when they began to

look at how organizations were organized, huge companies that have grown since the 60s, when the first investment in computers was made, software was purchased and big IT systems was the result, and these are still alive. The systems runs on different hardware, different operating systems, they are developed on different platforms and in different programming languages. There was no focus on integration between these, but the problem with integration directly arose when there were two systems. System integration is therefore no new concept.

From this environment where one has different systems there are islands of information spread around the systems, information is stored in different ways. The information is there but it is difficult to consolidate it, just because it is stored on different places and is structured in different ways, it can be hierarchical databases, file systems, relational databases etc. To create value out of this, one has to merge this to get the right information at the right time. Then there are other problems that have driven us into this way of thinking. People in the computer industry begin to retire and competences are lost concerning some products and systems, this will lead to future problems when one shall continue upgrading this systems.

The next picture is the technological changes on the market and they happen vary fast. One example is SMS; in the beginning SMS was built to support the operators in information sending between stations and there were no purpose that people should use this to send message between each other. Someone thought of this as a good idea and it becomes a technical innovation that no one thought of, this demanded rapid changes on the market. Organizations made use of this and started to sell ring tones, let people communicate with each other, no one hade predicted this. Suddenly through a technological change, a rather dramatically change on the market. Political decisions have also impact on the market, e.g. decision about the huge brand companies' monopoly concerning service of cars. The decision opened up for competitors. Furthermore there are fashion trends and customers behaviors that constantly changes he market.

If you take this two pictures; the complex IT environment and changes on the market, one realizes that if organizations what to meet competitors, changes etc. one has to change the business processes, and that quickly. Traditionally speaking, the picture of IT has been a guy in glasses who installs a software e.g. Windows on a computer. Nowadays one tries to lift up IT as a support system for business processes. To realize this and become on demand, one must have an architecture that supports rapid decisions and changes.

Before, in the time when organizations worked functionally, and saw IT as a "black box" and later went over and started to work object oriented, it was hard. One started to develop new classes instead of code reuse. The risk is that it will be the same with service oriented thinking, in the end an organization has e.g. ten services for the same task. The risk is that we build spaghetti but on another level.

4.1.1.4 Respondent D

I think that every new concept that IBM presents is "business a usual", we have used SOA for a long time or we have talked about it, but suddenly one wrap it to something that one can advertise and sell. But this time it is just not a marketing concept. My comprehension of SOA is that one develops modules and applications on basis of what the customer actually demands, in other word; which services the customer uses. This means that the customer buys the service, traditionally speaking, the customer buys capacity and thereafter services are designed. In this way one gets a lot of air in the investment. This is much like the previous "on demand" concept IBM presented a couple of years ago. You actually buy what you need.

4.1.1.5 Respondent E

Enterprise Resource Planning (ERP) systems are considered the heart in many organizations and it is critical if they break down, usually this cost a lot of money. Traditionally one have built solutions as information islands based on demands, e.g. HR, financial applications, order, warehouse etc. As times go by organizations have noticed that integration between these systems are necessary for cost effective business. If a customer places an order this creates a lot of background processes and when these processes execute and integrates successfully one have a complete value chain. This integration lowers the costs and e.g. shipment time which improves the organization as a whole and everything is done by one system. One example is Wal-Mart, if one product is taken from the shelf; a signal generates automatically to contact the manufacturer for production of a new product. Off course one does not produce just one product, but the principle exist.

This is the background for new architecture deployment. The question is if everyone needs to know everything about the different integration parts, the components in the landscape. Let's say a manager want statistics of the sale of products to take action for optimization. This manager should not need to log on several different systems to get this information; time is too expensive and should not be laid on in/out logging functions. SOA implemented at e.g. a medical company can be lift out and put into a steal manufacturer company. SOA is not magical, it is a way to do things and do things right and simple based on demand.

SOA emerged to simplify, e.g. if I need a service, any service e.g. an order service. My concern should not be on service providers, suppliers etc. I shall only subscribe on that service when I need it, at any time and at any place.

4.1.1.6 Respondent F

I have a general picture of SOA, that I believe I share with most people, but as far as a can see it is just talking, not many have implemented SOA as architectural form. On the field in my work I meet many customers and there is particularly one organization who have proceeded vary far in their SOA-work, IKEA. IKEA began with SOA architecture in the late 90th, and this was before Web Services and service busses. They wrote their own infrastructure and this was before XML and Java objects. The work IKEA did is now days called SOA architecture; they called it service duty orientation. My opinion is that today SOA is a "buzzword", for example; IBM, Oracle and Sun are three participants who pronounce it in this way. The all want a piece of this cake. My experience is that from hype to reality there is a distance of three to four years, I believe that if one wait, let say three years the hype is gone and organizations really begin to work service oriented. Now organizations are in the "testing phase", they ask themselves; what is this? What do we need? Which services can we develop?

4.1.1.7 Respondent G

It is a way of living, it is not a product, it is not software, it is a way of organize the development cycle and the systems within an organization. The systems should work as services rather then strong coupled components. On the other hand, I work with the infrastructure beneath SOA, the importance of a having a virtualized infrastructure that SOA is based upon.

4.1.2 What fundamental parts make SOA attractive to you?

4.1.2.1 Respondent A

From my point of view, in the type of business I work with, from a global perspective, forced through reports. Cost effective standardization of the reports are essential. It is also attractive for many companies who buy and sell other companies to integrate new organizations quickly and to do it cost effective. Maintenance of integration elements and application communication is very expensive for companies. Often do the huge global enterprises have different levels of maturity between countries, an organization can have bought some organizations, in one country they have come far but in another they are not so mature concerning applications and infrastructure. SOA is a cost effective strategic approach for consolidation of information.

4.1.2.2 Respondent B

I hope that the business and IT perspective will be able to talk to each other in a more natural manner. Both perspectives must lover the crossbar between each other. We have struggled with this several years.

Services must evolve from business perspective, not from IT; this is a very important factor. SOA is still just a buzzword, we have to make it visible, it has to be real and it is important that organizations can relate service oriented architecture to their business. It is still a lot of technological talk about the concept of SOA; the business perspective has not yet accepted it. I am afraid that if one does not let the business to be involved or if they do not take charge, it will fail and it is important that focus is on which services are essential and which clear business objectives are the reason why these services are created or used. Personally I think that this is very exciting, that business perspective and IT perspective are playing along together and when they do that we receive much better results and organizations reach business objectives much easier.

4.1.2.3 Respondent C

Reusability and quickness, those are two important parts too meet threats or opportunities on the market; one has opportunity to react fast and make quick changes in the organization.

Something that is in line with SOA is virtual companies. Today one can search the internet for suppliers, more or less successful. To make this faster and easier, SOA is a good approach, were one can build together elements of every step in the value chain and make money. SOA connects the transparent shell of the business functions.

4.1.2.4 Respondent D

Today IBM, like other outsourcing companies sell mainframe, UNIX based boxes, Intel based boxes and together with this the install some software and sell program for surveillance. This question is connected to the first question, if we sell services we are responsible for that service, the payment for the service and the facility we provide for the service. This means that the customer does not have to worry about what platform we use. This is an opening for the selling of SOA, because the customer does not need to know which services they need or which response time they want. I believe this can generate flexibility in our business and that it will be easier to make profit in our projects.

4.1.2.5 Respondent E

It depends on the organization and the objective, what do I want. SOA has no focus on technology. Compare e.g. Linux with Windows, windows, according to me became popular because of its simplicity and are still today because it is easy to use. It is simple to handle, one does not need to know what is happening behind the components. The simplicity is a fundamental part of SOA. The level of "on demand" or in other words; adjustment to the requirements, is important. For example, if I have a peak of labor in my work, in turn of the month, than I can order a service just for this occasion and receive it when I want.

4.1.2.6 Respondent F

The question is, is it for me or for my customers? I am a consultant and I help customers, what makes it attractive to me personally is that it will bring a lot of work. IBM has come far in their thinking of what SOA is and IBM has many products that support it, this in fact leads to education of others, our customers and we can help them a lot. It is always fun to take a closer look at new technology and new architectures.

What I believe customers can gain from SOA is the independence of platform, the independence in programming language and that client and server are completely independent of each other. It is also important that there are many different tools on the market to implement and develop their SOA architecture. E.g. if the customers have a business function that they want to web service enable, in principle one can with just a mouse click convert the service to a Java service or something like that. Development tools and the loose coupling are attractive and fundamental parts, and then there are some anxiety concerning the performance, for example if one wants to book a journey over the Internet, maybe one want to book hotel, flight and taxi, the travel agency will ask the airline company and they will ask the taxi company for matching taxis etc. This will create many different questions involving many different systems and it will be difficult too manage. The best way might be to choose a couple of partners to work with, and service level agreements.

4.1.2.7 Respondent G

The flexibility and the loose coupling is a fundamental part of SOA. As a hardware supplier you need to realize that systems demand much more power because the use and input of more and more XML and other things that demands resources within the systems. Along with the fact that things happen generally, organizations updates their systems environments and feels that there is an opportunity to do something. When an organization looks back at old stuff, they can get the turnover of the infrastructure rolling. It is also supposed to lead to a cost optimized infrastructure and more flexibility through the use of less, and the thinking that one need less.

4.1.3 What is your opinion were to start a SOA implementation and why?

4.1.3.1 *Respondent A*

It is important that one has, like in every project of change, support from the management lead, and often is these questions at CIO level. It is important that everyone understands that a change is going to happen, not just for the decision maker but for the connection to governance so that management can explain that this is the solution, but there must be business cases for backup. This is often the best way of working. Explaining that this will increase in higher quality and reduced costs motivates support from a strong governance strategy. For Swedish companies this is extra important because Swedish companies are more

decentralized than American. When talking about standardization it is fundamental to have strong governance around these questions.

4.1.3.2 Respondent B

It could be top-down, bottom-up or meet in the middle i.e. there are several ways to start a SOA implementation. Everybody involved in a SOA project must know their specific role and how to proceed. Huge efforts must be put in the beginning of the project. Most important is to first distinguish the business and their objectives, sure there are excellent techniques and people who will force the project forward and start with the actual program code, and one should not forget that all pieces must be included. One can start with bottom-up but must not forget the top perspective. It is important to map the business, one method that IBM uses, is Heatmap, a method for mapping all pieces of an organization. Break down the business and find the services.

One key picture organizations need to consider is that all that is made or done in the IT section must have a connection to the business and the pieces must be justified. With this, it will be easier for IT to get money and solve problems in a constructive manner. To create a solid foundation in the beginning is of most importance, not only in SOA projects.

4.1.3.3 Respondent C

The problem is that when presenting this to a company, they might have demands that shall give return on investment during a couple of years. It is then critical that one find a useful process to begin with and implement this. One can just no sell SOA as a component; it is architecture, a way of thinking. This is the strategic architectural thinking. IBM is one of the suppliers that can deliver a number of infrastructure components. To be successful with this one has to meet people in line of business that can be critical and someone that can calculate and present that it really works. This is essential all the way, for the customer it is important that he can accept the problem and recognize the market.

When one has implemented the thinking, it is critical to find a “test” process and begin with it and later move on. Does the customer find it good, and then we can build even further. It is only then we have implemented the thinking, a solid IT foundation and IT infrastructure. This can be as easy like connecting two systems with each other in a service based interface. The business side drives the development.

Why? If one shall survive in a world of rapid changes and with fast communications. With SOA it is much easier to compete, one can move into competitors markets because there are so many channels. Too meet changes, technological changes, political changes, one must have protection for everything that changes the market and service oriented architecture must support it.

SOA might bring changes in the culture, but it also demands grip of the technological infrastructure and the business processes, like one has to have sympathetic staff. Maybe we will se more and more dynamic organizational constellations, in the future one might not be hired and employed but everyone might be consultants who sell their competence and services.

4.1.3.4 Respondent D

The implementation has to start at the business level, it is always the business demands that carry on the business, no matter if it is SOA or not, I believe that it should be the business

demands that push the IT projects. It also has to do with how the IT infrastructure is designed? How does the application structure look like? If one start from scratch it is much simpler, but today there are few such companies. It is hard to tell what is right or wrong. If one carries out the project from the business side and form the functions the organization must fulfill and thereafter develop downward, this is the most logical procedure. First develop what the users need and to make the organization flexible, make it streamlined. The most important factor is to develop a solid architecture, after that the technology will follow, new technology will always arise.

4.1.3.5 Respondent E

I believe we shall say that one implements according to SOA, this makes it simpler to understand, because SOA is a way of doing things, a way of thinking, it is not a product. It is fundamental to think of SOA in terms and not as a product. Almost in every organization there are need for integration of applications and a great demand on system integration. Does the organization really need SOA? Smaller organizations might not make money with the help of SOA, the demand are just not that great, it should not cost more than it tastes. SOA makes more sense in larger companies, in bigger organizations with many systems and enterprise integrations. Philosophy speaking, SOA is a paradigm shift. From manually isolated information island and technology to a more service based approach.

4.1.3.6 Respondent F

Why should anyone use SOA at all? - Because SOA makes it much easier to build dynamic systems. The fundamental picture of SOA is that one shall develop small services and then use these services for further development to larger systems. If one needs to change a system, one only uses the existing services, and services should be stable.

Eclipse is one development tool and I think of this because I believe that SOA will be like eclipse, a development tool or a fundament where one can connect plug-ins which will provide functionality. These plug-in can talk close to each other and at that time I started to think that this is much like SOA. One will be able to plug in services and they will be able to talk to each other. Eclipse has had very huge impact the latest 2 and 3 years and one can draw a parallel to SOA, SOA might be huge in a couple of years. It will be much easier to rapid build applications, and this is maybe why one shall use SOA.

Where to begin has to do with existing services today or take a look at what the system demands. There are two approaches, one is “bottom-up”; evaluating existing services in the organization and which of these services are good for web service enablement. And second we have “top-down”; evaluating what applications the organization need to develop and what they require on basis of those developed services. My spontaneous reaction is that organizations work bottom-up, one start with existing assets and how to web service enable these to make them a part of a bigger process. This is the approach for technicians, but if you are business models your approach is top-down and start with your business process. Were to begin is related to your role in the company.

4.1.3.7 Respondent G

One should start small and well defined, business close processes that give payback within a reasonable period. The projects must be in the long term, not to complex, easy to implement, the first time it will probably be more difficult then everything that have been done before, this is very common when a new way of thinking is introduced, new techniques or new methods. The first one does is strenuous and will be slower in progress than the old way of

working, but in the end the organization will receive something better, the will improve their processes or something like that. Furthermore it must provide a better business result than earlier.

For example, if an organization does not have Enterprise Service Bus (ESB) and just buy an ESB for the large future project and tries to calculate the value of the smaller project in action, it might not work. Maybe they have to spread the cost of infrastructure between several projects. If they try to calculate the value of every investment concerning the first small project they might not succeed.

One important part in the SOA implementation process is the personal attitude; everybody must give the commitment to this, from business people to the developers who need to learn the new methods, to the people who implement it. The first time SOA is implemented one has to work through some inertia to receive a winning result.

The reason why one should use SOA is that one want to attack the stiffness in the systems today. There are different scales of the problem depending on the type of company. One problem area is concerned with the buying and selling of other companies, and in this way organization grows. Enclosed with the buying of a company one gets the connecting of new systems with existing, traditionally one has connected these systems with point-to-point solutions. For example, if a line is drawn between two bullets there is no problem, not even with a third bullet and a line is drawn to connect the three bullets, it is still simple. But when more and more lines and bullets occur, the paper fast turn into black and we have a problem. If one connect every system together in this way, with point-to-point connections, (product-to-product) figuratively speaking, we get balls of hair, also known as spaghetti structure. There is a driving force to work service oriented, to Web services, to Enterprise Service Bus, and especially loose coupling between sub systems. There is a thought in SOA itself and the processes around it that one should think of the business impact instead of focus on IT.

SOA allows organizations to lift out the implicit business processes out of the application and place them outside. If one want to change business processes one can do it without changes in the fundamental functions. The basic idea is that organizations should refine their processes and call upon the processes for support. This means that one very fast can change business models and meet threats and changes on the market, including the opportunities one get through doing business in a different way.

Much of the problem with today's IT environments is that here is not only the fundamental applications which do things, but here is also the business processes implicit coded. The people in the organization is in principle bound to what the applications do and in what order, in this way it is hard to improve processes.

4.1.4 What can we do to avoid the same mistake with spaghetti structure, building architectural spaghetti with Service Oriented Architecture?

4.1.4.1 Respondent A

I believe that solving spaghetti structure is the value of SOA, no harsh couplings between systems. Master data demands standardization, standardization of master data means that the information is defined in the same way; I believe this is one of the most strenuous parts. The labor does not lay in information sending, it is strenuous to secure the definition of information so that one has quality in the information flow and make it standardized. This

might be tricky, especially considering when every company in an enterprise not has implemented a SAP solution, which will make the negotiations simpler. One company can e.g. have sixty pay rolls, were one division uses excel and another division uses a complete different system, this is the huge challenge.

4.1.4.2 Respondent B

In the earlier days one could put a flap over the IT services, but now the risk is to prevent the problem from moving up to a higher level and services figuratively speaking get their own life and flies around in the organization and with this the whole organization have a problem. To solve this problem, organizations must open the box of IT and take one piece at a time. No big bang, search for small areas, learn from them, what is good and what is bad. Every organization is unique regarding to geography, culture, people etc. I think one need to do it for every little piece, but there are no obvious ways were to begin. One starts in a small scale and create comprehension and knowledge, the first project is a learning part for every one involved and with the next project one take a bigger piece, successive movement to benefit.

4.1.4.3 Respondent C

I have never seen that the services by them self should turn into spaghetti, but what I have seen is that the reuse and discovery of services can be a problem. This demand knowledge e.g. the people who work with Java must have huge knowledge in what Java libraries are available and which to use, so that one doesn't develop a new class that already exists. It is similar with services, there has to be a way to communicate which services exists within the company and which services to buy, there might be contracts with different suppliers of services etc. The services I refer to are services that are reflections of business processes. It is no difference if just one person knows which services the organization has, everyone involved has to know.

4.1.4.4 Respondent D

Weather it is SOA or not, a solid architecture are a good foundation for future problem solving. As long as one listen to the business' demands and let them lead and the management people stay in line with the agreed standards, and when participating companies join in one must decide if they fit and if they uses the same standards, do not let old standards remain.

If one takes a closer look at some huge companies, divisions have developed their own standards to fit their specific requirements and needs. In the end the company has everything on the market plus the internally developed. I believe this is a management question, some mechanisms of control. This is not only related to SOA, SOA is a tool for definition of business processes through the entire company. I think one can avoid architectural spaghetti if one let the business division manage in the start of the project.

4.1.4.5 Respondent E

This is a complicated question and there are no simple answers. One moves the level of complexity, from byte and bits and processes to a higher business level and functionality. If we start building far too specified solutions or services for specific objectives, spaghetti will appear, more or less code based. Instead of having ten simple portlets, we instead have two thousand portlets. We have then complexity at business level instead.

To avoid this one should think a little bit more enterprise wise, the reason of why we want the services. Which are the key functions or key services? Categorization of functions and processes, clear use cases. The responsibility as IT architect will be to manage what we have designed is useful and that we can use it. The technology, functionality or architecture must

not be an end in itself. We must not build architecture just for the architecture. If the organization asks for implementation of SOA and the organization is SOA enabled then I do not need to design a thousand different services for my solution, there should instead be ten to twenty large homogenous functional services that we have focus on in the architecture and after that simply write down scenarios.

4.1.4.6 Respondent F

Maybe one should have a good service registry for the technical part; further one demand control and management concerning projects. One need to evaluate what we have and what we can reuse instead of writing new applications, this is an important part to avoid spaghetti structures. Several persons are of the opinion that if they have not written it by them self it is not good because it is not invented here. This has really nothing to do with SOA; it has to do with management, communication and control of the projects. A good start is to have a service registry for orderliness.

4.1.4.7 Respondent G

It is up to the methods and people, how people comprehend the methods used and the problems one need to solve, today and tomorrow. But surely we will create new problems in solving old ones. It is just a part of the human brain how we think and how we solve the limitations we have, the system we have and the tools we have. Much of what we look at today and say; "how could we do it like that?" why didn't we do it like this? In those days there existed no tools, no maturity and there were other problems, e.g. if one has a slow computer one can not carry overhead of abstract software data and coding was made close to the bare iron. There existed no huge software packages and huge libraries for modeling or standards as XML. Without these attributes one could not build adequate solutions.

There is another problem, e.g. fashion, SOA is just fashion, and suddenly is everything SOA even if its not. A couple of years ago it was "on demand" and then suddenly everything was on demand, on demand coffee machines, on demand diskettes etc. the only thing one had to do was to put an etiquette on the object and it become on demand. The same thing can happen with SOA, that vendor put etiquettes on everything and call it SOA. There are vendors, companies that sell SOA services without knowing what they mean, this can lead to terrible architectures, but often these vendors are sorted out with the help of standards. This is another reason why to start small.

4.1.5 Who is responsible for a service during its life time?

4.1.5.1 Respondent A

It is up to the organization do decide, it is not up to IT. The IT divisions' responsibility as I see it is to provide support for information flow and that they work properly, but the organization has the final responsibility. If you take a specific process, e.g. data input from a customer, and then it is naturally that the people who work with it are responsible. I mean, IT is just a provider to the organization.

4.1.5.2 Respondent B

Different responsibilities according to organizational level, on the business level there are services that can be derived to a specific process, e.g. a credit control implemented as a SOA service. Because of this (business) perspective there must be some responsible for the service at this level. What is the service designed to do from a business perspective? Than there are underlying architectures, like the middleware layer, here someone must be responsible for the

service. The responsibility are also depending on the organizational structure, but there has to be responsibilities and ownership at every level in the organization, especially at the business level, otherwise one only move it to IT section and then there might be a problem, a similar problem that we have today with architectural spaghetti structures.

4.1.5.3 Respondent C

The customer is responsible, in worst case the support team. This is an organizational question, there are always different roles within a company that owns a business process, a business process does not exist by itself. The person who executes the service should have responsibility, in other word, the person who has developed the business process or the person who owns it. This might demand rethinking in the organization. If an organization has someone who "hosts" the service physically on a computer, maybe they own the service. But there are also the commercial side, someone sell a service which is a part of a larger business process, and of course everyone wants to own their own service. This is a question of business, organizational and law.

If you have a typical company, the responsibility will fall under line of business, and IT support the organization but they should not own the service.

4.1.5.4 Respondent D

Who is responsible for a service? This is a question of agreement. Form who do I buy the service? If a company buys a service from a service provider, it is then the service provider who is responsible for the service. The service provider is responsible for the service and within that, the application, the infrastructure, the power and network. It is possible that the service provider is responsible for all delivered components. These services can in fact be bought services from another provider.

IBM look at "grid computing" and as outsourcing supplier or provider we might be responsible for the deliverance of a specific service then we do not own the application or the hardware in use. But we are responsible for the quality of the service. It is actually simple, line of business buys a service from a service provider, and the service supplier is therefore responsible for the delivery of the service. But this is not obvious, traditionally speaking, that we as an outsourcing company are responsible towards customer for the IT delivery, one could have several different actors. If we do not know who is responsible for what, could most likely bring us back to the same spaghetti problem, this time with architectural spaghetti. Suppliers sell one or more services, a service can in fact contain several other services and services cooperate, someone must be responsible for the entirety. If one spread services between different suppliers, than the organization must have an architect who handles the structure to avoid spaghetti. This can be much disorganized.

4.1.5.5 Respondent E

If you think of a service, this service is surrounded by several layers. From bare iron to integration, in between we have databases, different types of data pools, but there is also an information bus were the services are transported; this is the huge integration bus for services. Of course there are users and the developers in the background. The responsibility for a service is actually shared, it has different owners or the different parts of a service have different owners.

For example, to make it simple, maybe to simple, the subscriber of a service or the buyer of a service is an authority who has responsibility for the service existence, concerning

maintenance and extensions of the service etc. We simply have a shared responsibility between actors and divisions.

4.1.5.6 Respondent F

During the development time it is always the development project that is responsible. When they are finished, the service will be moved to an administration unit, when the service is up and running the administration unit is responsible. The administration unit often needs help from development project team if there are changes to be. The problem is when the project is finished and the service or product is put into action, sometimes the project team is broken up and it can be difficult to reach the people who made the product or service, because these people might be involved in another projects. I don't believe this is specific for services; it can also be applied to products and other project types. During the development time of a product it is the development team that is responsible and later the responsibility is moved.

4.1.5.7 Respondent G

It is of importance that services, applications and composed applications which use services, have a life cycle. It has shown that it is easy to write an application but it can be torture to maintain it and above all, it might in the end be very difficult to get rid of it, just because no one knows what the service is developed to do or when it has finished. Life cycle management is very important when it comes to handling of services, in principle they have to be treated as applications, much like product life cycle (PLC).

One must not forget the (master) data that is the foundation of a service; the data is often the forgotten child in SOA. If one has a high level of abstraction concerning a service, there are always some data sources which are used to create this high level. Than one needs to know if one can trust the data in the source, and who, if it is a customer register which are built up of three earlier customer registers because the buying of several companies, how does one know if this customer register is better then the earlier? If there are three customers called John Smith, is he the same person in every system or is it three different persons? Is it the same person, but with different addresses? How does one secure good data? For example, if you mix, in this case customer register with something else, you get garbage. Much like mixing ice cream with fertilizer, all you get is fertilizer.

4.1.6 What information do you need to know about each service? On what basis do you manage identified service attributes?

4.1.6.1 Respondent A

The importance of the governance questions is to push for ownership around what we call master data. If you take large system implementations e.g. a SAP implementation and if you do it by the book you have to put a lot of effort into securing master data and that they are defined in the right way. Have you done this properly and apply your solution the processes, applications and other solutions will follow. It is when no proper definition of master data definition exists, no proper handling of processes and no proper work definition around this questions problems will arise.
Master data has a strong connection to the system of reports and what do you want to get out of the system that is important and drives the demands. If you want to do this right, IT should build a model that satisfies the demands that the enterprise has on reports and quality of master data and make sure that an application is available for those demands. It is also a question of governance. Some companies who force the governance question to the limit centralize their entire IT budget, IBM is one of those companies, and divisions will not

receive money to development of applications without first going through governance. In some way they take the money away from countries and IT divisions and store them centrally to secure that the way of working. It is important to control the right things, I believe that the level of governance should be high concerning IT, but governance can only be high leveled if one has succeed and made it right, it will be more difficult if the development is wrong. That is way it is important when you drive these projects that the business side is represented and that they have requirements from the beginning.

4.1.6.2 Respondent B

To this I have no perfect answer. In the future there will be many different services; business services, technical services in many different shapes, therefore it will vary. When it comes to business services one need to know which roles there are, what the organizational requirements are, if there are external participants, are there agreements involved, if there are changes in the market to expect e.g. governmental rules and regulations such as the law of personal records.

One of the fundamental pieces for this question is to know what you need to know for supplying a specific service, the requirements of the service, what input and output should the service provide and simultaneously someone who delivers this and then we have created a responsibility matrix in some way so you can see which part is owned by whom. One must write down and decide depending on the structure of the organization and which competence is available in the organization. Controlling the intellectual assets is important so one doesn't develop solutions that already exist. It is important to do this procedure on every service to get all the information required of the service.

Service oriented architecture is a new paradigm and there will not be many services in the beginning, but after a while there might be hundreds and then it is very important not to invent the wheel again, that the organization has control over their services. Control should be a part of every implementation and every project; it must be a natural part to have a reference library where services are collected. A reference library (repository) towards the business with business services and a library connected to the middleware and so on, this is a part of the whole SOA model.

4.1.6.3 Respondent C

It depends on who asks the question, as a developer one need to know where to find the services, based on some attributes. (IP address, name, interface, protocol) One also has to know what different functions the service provides and in-put and out-put parameters.

The business side uses technological definitions of a service, but instead they should be concerned with what the service provides and what to put into the service to get the best result. The information one need to know about a service is different depending on who asks the question. The demand of the business is prior to the demands of IT; IT will receive a requirement definition of the services needed.

4.1.6.4 Respondent D

It depends, in my role as infrastructure supplier I need one kind of information and I need comprehension of desired transaction quantity. This one can get from e.g. different orders, invoices etc. I also need to know accessibility and response time, is there need for service maintenance or should the service be up and running twenty for seven. Just like the supplier of the application who needs to know what and how to design the service. There is actually no

difference from how we do today. I need to know the impact on the business if the system should crash and prioritize in problem situations. Geographical information of the users is also essential.

4.1.6.5 Respondent E

It depends on the life cycle stages. E.g. in the development stage one need to consider what do I need for this service. The information about each service will vary depending on the stage, the design stage, implementation stage and operation stage. But most important, the classic fundamental part in architectural contexts are the objective, why shall we have this service, what function does it provide. What is our need? To satisfy the organizational need we must know what the actual need is, what is the objective, this is essential. When the organization knows this, the rest will follow. One also have to ask, is it worth it? Or is there another way to do this? Or do a really need this service, what is the benefit.

How to make services visible is connected to the organizational structure and the culture, how does the organization work, how are organizational questions handled. To fully take benefit of SOA and to market SOA in a good manner so that it make sense to everyone, or at least those who need it, one have to look at how the enterprise is designed and how it works today. Does an organization have old habits and therefore will new ideas be treated as crap. The organization really need to consider what the need and what can help them to fulfill there goals, look at the demands once again. Never base SOA on obligations, many companies are based on obligations, it is much simpler to consider the requirements instead. On demand, access based on need, the way of working should be moved to service thinking instead of individual needs.

How to manage services in a company, one must concentrate on one's need, not on IT or the end in itself, instead focus on functions. Until now, the world of enterprises has focused on business units instead of market segments. Suppose you have a number of enterprise areas within a company or number of different companies or maybe different sectors. What I believe is that one think of one's services according to the distribution sector. One has to market the services to develop awareness of the services.

4.1.6.6 Respondent F

If you develop a project and use a service, or if you need a service, the first thing to do is to define which services are needed, which are the demands on the service provider, what is the cost, answering time, availability, security etc? What service level agreements does the service contain? It is important that one dare to use the service, but you also need to know that it is the right service.

If you shall use a service you must secure that the service upholds the expectations, demands and which functionality it provides. You also need to know where it is and how you connect to it. Furthermore you need to have fact about who uses it and who has used it, if you are supposed to charge the consumer every time he/she uses it or by data etc. Different business models decide this, as provider I must manage this to charge the right fee for the genuine service.

4.1.6.7 Respondent G

The respondent prefers not to answer this question it is too thorough to answer in this short amount of time, as well as it is not in line with his competence area.

4.1.7 How can we spread information about service existence to other people in the organization?

4.1.7.1 Respondent A

People say that this will become better; no one likes to work in vain. You sell a high-quality solution that provides the information you need and you make it work, the flow of information is managed cost effective. There is no guarantee that the solution will work hundred percent but if you can achieve eighty percent for most of the countries and organizations, then we can make adjustments locally where we need but the foundation shall be the same. Information dissemination, I do not believe in loose control, if you say something like "this is good, take a glance at it" if you have this attitude, it will most likely not succeed. It is better to say "I have something here that will meet your requirements to eighty percent". One must sell and sell based on advantage with the solution, this is comfortable and this is good and it works. It is of enormous importance that you do it right from the beginning. Not every company wants to spend this amount of money and time, if organizations are careless in the beginning and receives a lower initial cost, with this it will be more costly the longer you precede. Many times it is better to take a large investment in the beginning; this will result in lower cost in the future. This is how IBM functions.

4.1.7.2 Respondent B

One way is to apply a service catalogue at a portal in the organization, were one handles the services. If you look at e.g. the financial side of the organization you can for example possess twenty services and they deliver a value based upon a certain input etc, in this way you have a description of every single service. Then we must find the appropriate description level that the entire organization can understand, it is important be pedagogical. The description must not be too simple, then it might mirror to less but it must not be too hard, a balance between them for best understanding and knowledge of what it means. Everything is of course dependent on the type of organization, structure, culture etc, and which competence level the organization has. Most important is that the information about services is accessible and not hidden.

4.1.7.3 Respondent C

One need to have a data layer in which services are well defined, then one need to ask if everyone need to have access to the service. I suggests "*Web Sphere Portal Server"* one of IBM's products, this gives you role based access to the information. The product is excellent to front an information layer; it is the information that should be presented in one way of another. How this is best defined is probably something that a behavior psychologist and character psychologist can investigate. But the information must be presented some place within the company and "Web Sphere Portal Server" supports this.

4.1.7.4 Respondent D

One should have some kind of decision forum were both business and IT architects are represented. I do not believe that one can spread information through an intranet. I believe in "rock hard" governance from the management, it is the only way.

4.1.7.5 Respondent E

Simply by trying to change the behavior and culture of work/current processes in a organization and shifting the mindset of the individuals, by introducing new ways of acting and thinking efficiently, would basically increase a 'natural' need/demand/requirement for new services.

4.1.7.6 Respondent F

Today email works fine, but this might lead to spaghetti and handling of versions. This is not specific to services but the same thing concerns how one disseminate information about a system or a specific project. If one look at huge organizations or let say, internet, I believe that service registry is a good way to begin, and that it is flexible so that one can search for the services. There must be good search engines to find services. There are UDDI;s that acts like public registries. The reason why I speak of this that I needed a service to send SMS from a website to cell phones. I found many different services but no service worked, services which correspond to my demands didn't work, public UDDI;s does not seem to work according to me.

I believe in service registries which are to easy search through for services, both in words and in parameters, this is important. Service registry is one way to spread information, much like information for telephone numbers; they market their service to be visible. In the same way organizations shall advertise their service registry to internal projects.

4.1.7.7 Respondent G

There are much talk about the movement of application development to the hands of the organizations, not normal coding but combined coding, much like a spreadsheet, were data can be picked form different services (spreadsheets) and be combined together. How the publishing of available services should be carried out, I do not know. If it should be in an environment with this kind of tool, one should publish within the tool which type of services there are in the organization. There should be some measurement of quality of data so that one doesn't mix ice cream with fertilizer. There should also be some policy decision.

There are companies which allows a big part of the company to be driven in excel, one has some kind of parallel processes were the spreadsheet is used as input e.g. what to perform a specific year, and then the next person in the chain fill in his/her stuff in the spreadsheet, this will lead to spreadsheets all over the organization. And there are companies were this is strictly forbidden, everything that have connection to the company has to be driven within an existing ERP system.

The application of services is developed by them whom are pointed to be experts on the subject, e.g. a computer division containing consultants, well defined projects. Or one releases the stuff and let people combine services with the new development tools that emerge or worse, write something in JavaScript. If one chooses the second way, one either has to publish in a more understandable manner, but if one chooses the first path one does not have to publish which services exists within the organization, it is then enough that the experts who are told to do this type of development knows.

Concerning Service Registry, to have a service registry will become natural, I do not know exactly which kind of service registry or at which level it should be. For some service registries it is of most importance that services can find each other. It is not necessary of the reason that someone need to use a service or at high level develop a service or comprehend what the services actually do or which processes the services belong to or what the reason of the service were in the beginning.

4.1.8 What basic building blocks are involved in an adequate governance strategy for SOA?

4.1.8.1 Respondent A

There are two main components; one is governance around the applied technical architecture, do we use the right products, decisions around e.g. technical platform etc. Second, governance concerning master data is of importance, there has to be strong governance around how one handles master data. This is often a functional question, often have organizations owner concerning HR, owner or financials, for supply chain, one for product development, those owners drives the ownership concerning master data. This is much more strenuous than the technical architecture; it is simpler for companies to say; now we only work with Web Sphere as integration platform.

Governance also holds the investment costs, controlling them, this is a way of securing the principles one want to use. It is difficult but it is important to stretch one's "tentacles" far into the organization. It is also a matter of education, securing the correct competence and that the employees have the right knowledge of how the organization wants to work. Furthermore education and securing of competence lies at governance level. These four components; technical architecture, master data, finances and education should be driven inside a governance function.

4.1.8.2 Respondent B

Governance, like I said before, there must be a foundation and one must sell the whole belief of SOA. To sell the belief one must start with a pilot project were everyone can be involved and feel that they participates, the organization must see that they gain value from this new way of thinking, after this one can publish the whole idea with SOA. One start with small pieces and simultaneously learn something on the road, through this the organization hopefully sees how the different roles in the business division can interact with the IT division. Governance is the framework and can contain a lot. If the organization can succeed in their governance work the rest will follow. Technically there will be several alternative to do this, but does the organization know what they want to accomplish, why and everyone have given their commitment, then the identified problems will be solved in every project. But I think an initial journey must be made. I do not think one can expect that organizations can absorb much from larger projects and understand every role, manage talking about services and identification of applications.

SOA is nothing new, from an architectural perspective we have talked about it for approximately ten years, the new thing about SOA is the reusable programming code and that we shall have services, object orientation is pretty close to SOA.

4.1.8.3 Respondent C

SOA governance is an important part, so are management and control, it is also a strategy were to head and why, the foundation might be that one want a rapid organization that can meet changes on the market. Governance is of course a very important bullet because it is here one might do the same mistakes as with object oriented design. Another issue that is important is security. The motive when one first started to talk about Web services and why it did not become so successful; is that organizations just thought that they could sell Web services as a product out of a commercial perspective. This never went through because the commercial part and the security part were never solved, one could not sell it because there was no support for it on the web and it was hard to advertise.

I believe that there are services that not everyone should have access to. It can be a service that someone has, where the cost is accordingly big and one does not want anyone else to use it, e.g. a supercomputer where every clock cycle cost a lot of money and this is needed to make financial analysis which are a part of a business process that in turn is a decision process for the management lead. It might be that the organization does not want to share these processes, because we might have a company with internal invoices and this can lead to access limitations.

Of course a SOA strategy also includes an implementation plan; you don't make a SOA implementation in a day. It will take much time; in the strategy has goals and objectives, and to achieve these goals one has tactics, tactical decisions to reach the goals. These can change over time depending on context e.g. changes on the market, buying of a company, selling of company divisions that influences tactic and strategy.

4.1.8.4 Respondent D

Organizations need clear business requirements, how to get this is difficult to say. But the organization always has business strategies and business architecture. Hopefully the organization knows why they are on the market, if they have a clear definition of this it will lead to better decisions around the IT architecture. It is mostly hard for the IT division to adopt why the organization exists. On the basis of the business architecture, one needs to define which support is needed to be competitive. A clear business plan is essential and I believe most of today's organizations have that, but I believe that the business division does not communicate with the IT division in terms of business demands. One just turns to the IT division and requests a database etc. There is actually not enough pressure on the IT organization. One sees them as a technology supplier and when there are problems or the costs have reach its limit, the business starts to ask questions. The tricky thing in all this is the communication, and there has the IT architect a huge responsibility to translate the business demands into a comprehensive language.

SOA helps out with structuring organizations many levels, from the management to the applications, sometimes it is fine sometimes it is not. If one talk to the highest management people there are no clouds but one must probably use a business architect who put together the process for definition of business processes and then convert the processes to IT terms, here we have the IT architect. Close collaboration between the levels and communication is essential.

4.1.8.5 Respondent E

Again use cases, the usefulness and demands are essential. Of course there has to be an IT infrastructure. Different organizations think different when it comes to SOA, e.g. SAP does not think SOA, instead they think ESA - enterprise service architecture, the foundation is the same, move away from IT as an mean to an end and instead focus on the requirements. It is just different names, e.g. IDI – IBM dynamic infrastructure. One can also think of "adapting computing" it is also a buzzword right now.

Consider a room, in one corner we have bare iron, (Hardware, discs, "the brain" etc), and in one corner we have data storage, metadata, in the third corner we have network, communication and integration and in the last corner we have the functions (the services needed). How do I connect these four corners so that I can connect anything based on which

demand I have? The fundamental parts according to me for SOA are; infrastructure and some kind of need and purpose, some realization layer and an enterprise integration bus.

SOA governance is a wide concept. In IBM's IDI there are standards and shapes to manage this (governance) surveillance, handle, automate and make it dynamically useful. Automatically and dynamically are different things depending on definition. Governance is one part of this, and of course must there be rules and regulations. Otherwise one might move the complexity up to another level and closer to the user and we have a new problem, similar to the old ones.

4.1.8.6 Respondent F

I do not know SOA governance, but I believe that this is where organizations want to begin, to start in the accurate corner with control and management. But take a look at internet, how did Google start? - Two guys in a garage. How did Bill Gates begin? - In a garage. But take a look at UDDI, it begun with control, management by rules and regulations how to develop and use. But this is an example that it does not work and no one uses it. Google, on the other hand is used by many.

I am not convinced that one shall have management and control from above in a company, however, companies want to have control over their assets, but I believe that in many cases this can impede development. Take a look at Google again, it is a company that has been very successful in this years. What I have red is that they work in a little bit different way, four days a week they work in compounded projects but the fifth day people can arrange their own projects. I do not know if this is the truth for everyone, but most of Google's employees have their own hobby projects parallel to their work, this is how Google Earth was developed and became product. I am a little bit of a rebellion when it comes to management; I believe it can hold back development.

Governance in projects is good, that is true, but because SOA is a new phenomena one might not know how to manage. One does not have enough experience concerning SOA, after some years it might be easier to control and manage. It is not unusual that new names concerning old stuff emerge in the computer industry, service oriented architecture is in fact nothing new. Every time something new are about to be released, it will be more adjusted and improved to fit into present day, but I do not know if I can call it fundamental news. The name SOA is new, but one must first learn more about it before one start to manage.

4.1.8.7 Respondent G

Knowledge and comprehension, learning by doing in small projects, it is very important that the business division is involved. It is of importance that one work with open standards even if the company should choose to implement SOA in locked architectures from a vendor e.g. one thing one want to address is the ability to fusion companies and then there is an opportunity that one doesn't use the same vendor and one get trouble.

One need to determine which standard to use, way of thinking or methods for life cycle management of services and that ownership of services are well defined. It is very important to have a virtualized infrastructure, an on demand infrastructure.

Many people talk about the importance of SOA governance, that it is important to include SOA governance in the organizational strategy, but what SOA governance should contain, that I do not know.

5 ANALYSIS AND DISCUSSION

In this chapter we analyze and discuss our theoretical frame of references and our case study. As we stated and explained earlier (see 3.1.3.3) we present the analysis by categorize the respondent's professional position and the questions in tree respectively five categories. The picture below illustrates the course of action for categorization of respondent's professional position and the line of action regarding achieving conclusions:

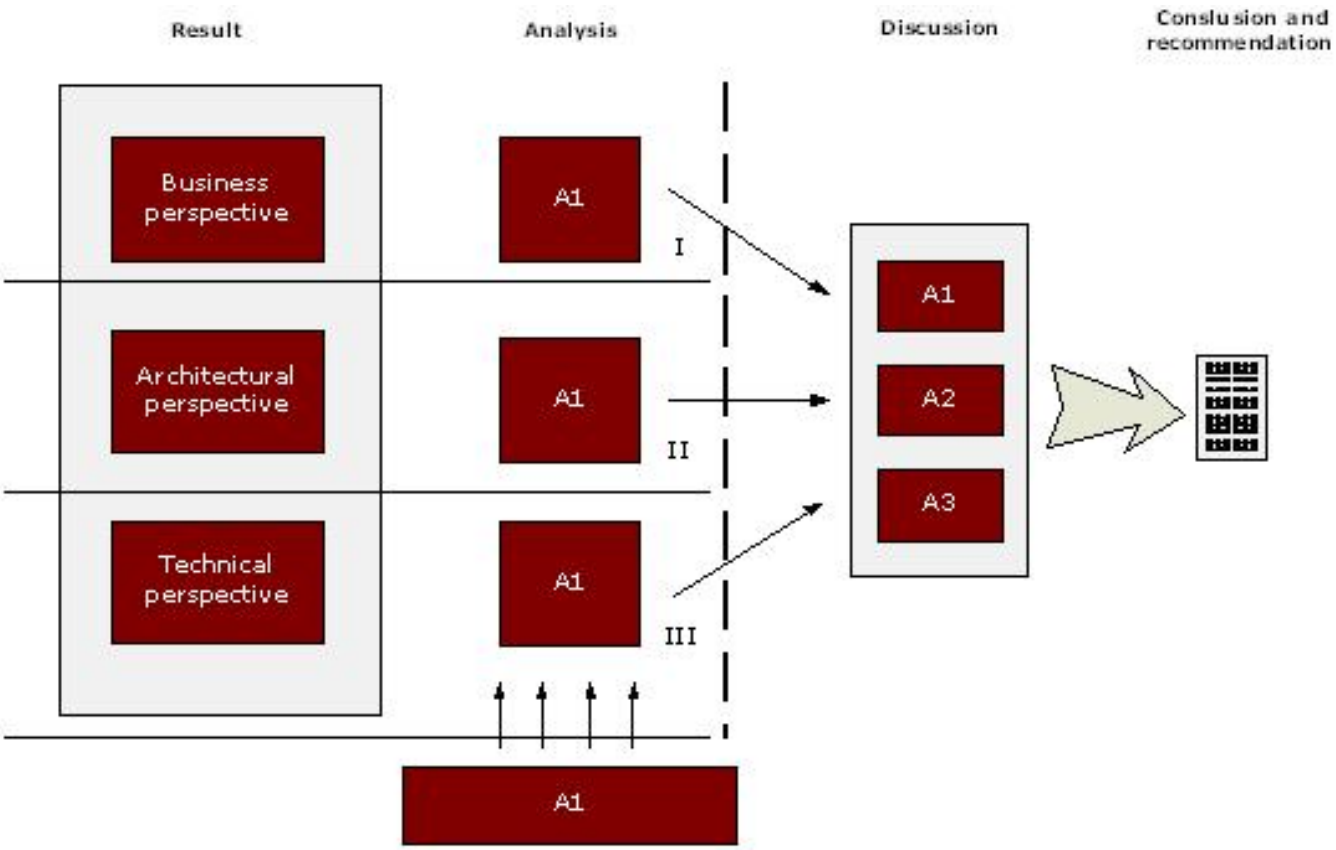


Figure 12. From result to conclusion

How questions were categorized is demonstrate in picture below;

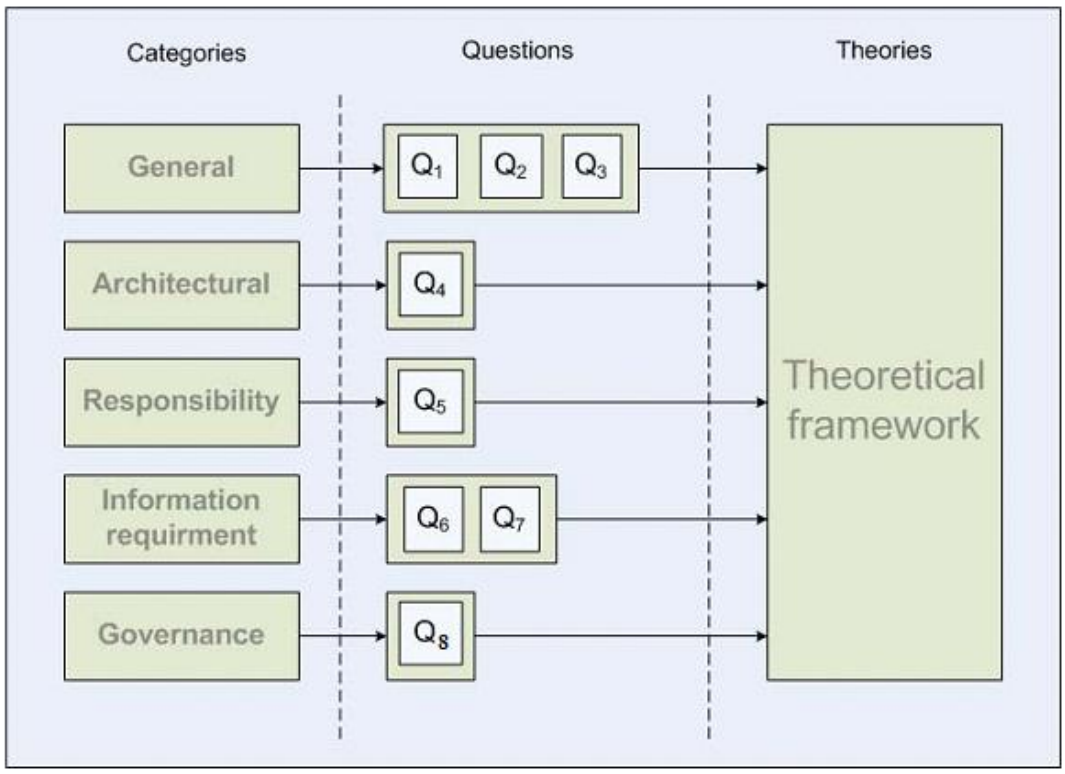


Figure 13. Categorization of interview questions

5.1.1 General questions

5.1.1.1 Business perspective level

The respondents in the business perspective level says that SOA is a concept that aims to create flexibility and that it is a cost effective way to integrate different applications, which will make it easier to maintain and connect new and old applications, and to standardize the management of the information architecture. It is also an architecture that better meets the forces on the market. To meet the changes on the market one must have an underlying system that supports changes in the IT system; SOA is the architecture which realizes those rapid changes.

The case study show that SOA out of a business perspective is all about business services; it is not technical IT solutions that build up services and creates the infrastructure of an organization. From a global perspective forced through the demand of reports, cost effective standardization of the reports are essential. SOA is a smart way for companies to integrate new organizations quickly and to do it cost effective and SOA is also a cost effective strategic approach for consolidation of information. We agree with the respondents when they say that services must evolve from business perspective, not from IT; this is a very important factor. We are also in line with the respondent that the business perspective has not yet accepted it, because they say and we understand that there are still a lot of technical discussions around SOA, the technical terms do not address line of business and we are still facing the problem with communication between the business division and the IT division. The IT industry has

struggled with this several years. Positively will line of business and IT perspective be able to talk to each other in a more natural manner and bring the cap between each other. Both perspectives must lower the crossbar between each other and we believe that IT management and SOA governance along with IT governance can bring this to reality.

The case study show that making business processes faster and easier, SOA is a good approach; building together elements of every step in the value chain to make money. SOA connects the transparent shell of the business functions. Several times when we reviewed the theoretical material and the case study important statement and key words come to our attentions. The respondents often talk of two fundamental parts of SOA that makes it attractive; reusability and quickness, it is important to meet both threats and opportunities on the market appropriately. With SOA one has opportunity to react fast and make quick changes in the organization. Windley (2006) say if there is no control of projects or no one that influence people to adopt SOA and service reusability as a fundamental part of design, then projects will go their own way, driven by requirements of a particular business unit or project.

Were to start a SOA implementation slightly differs between believes in the theoretical and the case study. It could development methods such as top-down, bottom-up or meet in the middle; there are several ways to start a SOA implementation.

From our gathered material based on case study we see that service oriented architecture can not be implemented as a product or a component; it is architecture, a way of thinking. This is the strategic architectural thinking, most important is to first distinguish the business and their objectives. SOA might bring changes in the enterprise culture, it also demands grip of the technological infrastructure and the business processes, equally a sympathetic staff and support from the management lead. Questions concerning implementing a new architecture are often taken by CIO. It is fundamental to have strong governance around these questions. Erl (2005) state that to prevent organizations and IT from being overwhelmed by this new complexity, an organization need processes that make serious decisions about requirements, design, implementation and maintenance which results in rules to increase control, overview and awareness. We agree with Erl and point out that to manage an SOA implementation, which actually from our point of view and some of the respondents too are a paradigm change, demands governance and strategies.

5.1.1.2 Architectural perspective

Answers and opinions from the architectural perspective do not differ much from the business perspective. SOA helps develop modules and applications on basis of what the customer actually demands, in other words, what services the customer needs to use. With this, it means that the customer buys the service, usually speaking, the customer buys ability and thereafter the needed services are designed. We see similarity with the respondents reply at business level, that services must evolve from business perspective, not from IT; this is a very important factor. SOA is not magical, it is a way to do things and do things right and simple based on demand.

Enterprise Resource Planning (ERP) systems are considered the heart in many organizations and it is critical if they stop, usually this cost a lot of money. According to Earl (1989), Ward et al. (1990) and Brancheau (1997) are several corporations dependent on administrative systems to be able to manage their daily activities. IT-systems influence many questions outside data-management such as organization of work, economy and inter-organizational co-operation. The respondents say that organizations have noticed that integration between these

systems is necessary for cost effective business. When processes perform and integrates productively one has a complete value chain. This integration lowers costs and e.g. shipment time which improves the organization as a whole, and everything is done by one system. Philosophy speaking, SOA is a paradigm shift. From manually isolated information island and technology to a more service based approach.

Today outsourcing companies sell mainframe, it could be UNIX based boxes, Intel based boxes and together with this they (the vendors) install software and sell program for surveillance. If companies sell services, then they are most likely responsible for belonging services, e.g. payment for the service and provision of extra resources for the service etc. This means that the customer does not need to worry about which platform in use, or which programming language the software is written in. This is an opening for the advertisement of SOA, because the customer requires only knowing which services they need or which response time they want. Simplicity is a fundamental part of SOA.

Besides simplicity, it is fundamental to think of SOA in terms and words, not as a product. This is a statement that often appears in the theoretical frame of references as well as in interviews with the respondents. We can not enough emphasize that SOA is not a product it is a way of thinking and doing, it is more a method than nothing else. Almost in every organization there is need for integration of applications and a great demand on system integration. Our case study tells us that smaller organizations might not generate money with the help of SOA, the need for flexibility, loose coupling, reusability etc. in smaller companies are just not that great, it should not cost more than it tastes. SOA makes more sense in larger companies, in bigger organizations with many systems and enterprise integrations, companies who buy other companies, and so on.

Many times during our study the respondents say that the procedure of implementing SOA has to start at the business level, we agree with this because it is always the business demands that steer the business, no matter if it is SOA or not. Therefore, it should be the company's demands that push the IT projects. It also engages how the IT infrastructure is designed and how application structures appear. Starting from scratch is simpler, but today there are few such companies. The most important feature is to develop a solid architecture, if one is successful with this, then the technology will follow, and new technologies will always arise but a paradigm does not shift that easily.

According to the Grembergen HICSS conference (2006) IT governance can be illustrated as the objective to form simplicity about decision rights and answerability, to create a sufficient level of desirable behaviors within and among organization s in a complex enterprise. We believe that this is useful when implementing service oriented architecture. IT governance aim to create simplicity about decision rights and accountability. Furthermore to create an adequate level of desirable behaviors in complex global organizations that varies in every enterprise.

5.1.1.3 Technical perspective

As stated before SOA is a way of living, it is not a product, it is not software, and it is a way of organizing the development cycle and the systems within an organization. The respondents on the technical level also agree with this. The systems should work as services rather then strong coupled components. On the other hand, you should not underestimate the importance of having a virtualized infrastructure that SOA is build upon. As the theory enlighten, IT infrastructure can help the enterprise, with different methods, to inspect which services they

offer. In this way the organization can determine which services should be shared external and which should be internal. (Ross & Weill, 2004)

The respondents' states that SOA is today an advertising word, e.g. IBM, Oracle and Sun are three participants who pronounce SOA it in this way. The all want a piece of the cake, wait approximately three years and the hype is washed away and organizations really begin to work service oriented. We believe that this is because SOA sometimes is market as a product, and of course suppliers of IT want to sell, but the problem is that SOA, market as a product, do not solve the problems with integration and spaghetti structures, it is not a magic wand you can place in the enterprise and the problems are solved.

Once again the case study show that the fundamental parts that makes SOA attractive is the flexibility and the loose coupling. Also, that there is a belief in SOA itself and the processes around it, think of the business impact instead of focus on IT. SOA is also supposed to lead to a cost optimized infrastructure and more flexibility. Furthermore, important parameters are development tools, independence of platform, the independence in programming language and that client and server are completely independent of each other. We think that one of the values of SOA is to attack the stiffness in the systems today. SOA allows organizations to lift out the implicit business processes out of the application. The basic idea is to refine processes and call the processes for support; organizations can change business models rapid and meet threats and changes on the market, including the opportunities that arise by doing business in a different way.

There are different scales of the problem depending on the nature of the company. One problem area is concerned with the buying and selling of other companies. With the buy of companies, the organization gets the problem and complexity concerning connection and integration of new systems with existing systems; traditionally companies connect these systems with point-to-point solutions. We agree with Checkland (1981) and Magoulas and Pessi (1998) that it is most significant to take IT management in consideration regarding business integration. IT management support business and application integration projects, and include handling problems that deals with applying new technology, development of new information systems, and maintenance of existing investments in IT.

The technical respondents' thought of the implementation as to start small and with high quality definitions of what is needed; this is what they recommend when implementing SOA. Identify business close processes that give payback within a reasonable period. The projects must be in the long term, easy to implement and not excessively complex. Implementing SOA for the first time will probably be more difficult than everything that has been done before; this is common when a new way of thinking is introduced, new techniques or new methods. One important part in the SOA implementation process is the personal attitude; everybody must give the commitment to this, from business people to the developers who need to learn the new methods, to the people who implement it. The first time SOA is implemented, one has to work through some inertia to receive a winning result.

Furthermore, to begin an implementation has also to do with existing services and focus on system demands. The technical respondents also state in unity with business perspective level that there are two approaches; one is bottom-up, looking at existing services in the organization and which services are suitable for web service enablement. The second approach is top-down, what applications do the organization need to develop. From a technical point of view, organizations work bottom-up, starting with existing assets and how

to web service enable these assets, moreover to make them a part of a bigger process. If you are a business models your approach is most likely top-down and begins with business processes. The respondents at all perspectives agrees on that were to start an implementation is related to the professional role in the company, this is essential to our thesis as well, like we stated early we aim to widen the knowledge about SOA and study it in different dimensions within an enterprise.

5.1.2 Architectural questions

5.1.2.1 Business perspective level

In our introduction and theoretical frame of reference we bring attention to integration and problems that organizations face when integrating existing systems with new systems, the problems are spaghetti structures and information islands. As we wrote in the beginning of this thesis Magoulas and Pessi (1998) say that this can also be defined by separated information systems distinguished by duplicated information in an anarchistic way.

Respondents at business level had a similar opinion about architectural or service spaghetti and they do not believe or have not seen that the services by them self could turn into spaghetti, but instead the reuse and discovery of services can be a problem. This demands knowledge e.g. people who work with Java must have huge knowledge in what Java libraries are available and which to use, so that one doesn't develop a new class that already exists. It is similar with services, because services are reflections of business processes. There has to be a way to communicate, exactly which services exist within the company, which services should be bought and there might be contracts with different suppliers regarding services etc.

We think that to gain value from SOA we have to avoid the same mistakes we have done earlier; building spaghetti structures, and further more, move the problem up one level, to architectural service spaghetti. We believe that this can be solved by control and managing of services and having good governance. According to Sifonis and Goldberg (1999) the demand for governance are based on the growing demand for attention to corporate responsibility concerning management and integration of the service payment to organizational performance, thus creating an rising demand on effective IS/IT systems. We stated this before but it is worth saying again, it is truly important to bridging the gap between IT and business and improving the communication.

Regarding solving the spaghetti the respondents believe that organizations must open the box of IT and take one piece at a time. No big bang solutions, search for small areas, learn from them, what is good what is bad. Every organization is unique regarding to geography, culture, people etc therefore there are no obvious approaches were to begin. One way is to start on a small scale and create comprehension and knowledge. The first project is a learning part for every person involved and with the next project one takes a bigger piece, successive movements to benefit.

5.1.2.2 Architectural perspective

All respondents agree that solving spaghetti structure is a complicated issue and there are no simple answers. If the level of complexity moves from processes, byte and bits to a higher business level and functionality level, control and well designed structures of information is essential. Also, if individual solutions or services for specific objectives increase in number; spaghetti will appear, more or less code based. Instead of having ten simple portlets, we have two thousand portlets, and then complexity will instead become visible at higher level.

From the theory we clearly se that weather it is SOA or not, a solid architecture is a good foundation for future problem solving, several respondents state that and we surely agree. As long as attention are given to the business and letting them lead, with this people at management level adjusts to the agreed standards a SOA implementation will become easier.

Taking a closer look at some huge companies, internal divisions have developed their own standards to suit their specific requirements and needs. In the end the company has everything on the market plus the internally developed. This problem is a management question, there has to be some mechanisms to control and manage IT. This is not just related to SOA, SOA is a tool for definition of business processes through the entire company. One respondent state that to avoid architectural spaghetti; let the business division manage in the start of the project. Once again, we think that controlling services and having good governance is the problem solver, also communication between managers improves organizational performance, and a manager does not have to understand how his/hers information system works, only how to use it (Ackoff, 1967).

5.1.2.3 Technical perspective

Respondents that go with a technical perspective, say that looking back on history generates questions, e.g. “how could we do it like that? Why didn’t we do it like this?” Elderly systems have a tendency to bring problems when organizations makes upgrades in business activities and IT environment (Galliers & Leidner, 2003). The absent of existing tools, no maturity and the non existence of huge software packages and huge libraries for modeling or standards as e.g. XML. Without these attributes one couldn’t build adequate solutions. Technical developers need to look of today and see what can be reused, instead of writing new applications; this is an important part to avoid spaghetti structures. For example, many individuals are of the opinion that if it is not invented here and by “myself”, it is no good. This has to do with management, communication and control of the projects. A good start is to have a service registry for orderliness.

Also the technical perspective gives attention to the importance of control and management concerning projects. One answer could be to include a good service registry for the technical part. Further more; to avoid spaghetti structure is up to the methods and the people, how people comprehend the methods used and the problems one need to solve, today and tomorrow. But surely one will create new problems in solving old ones. It is just a part of the human brain how we think and how we solve the limitations we have, the system we have and the existing tools.

Something we think is alarming, is that one respondent says that there are vendors, companies that sell SOA services without knowing what it is; this can lead to terrible architectures. We stated before that SOA sometimes is advertised as a product, not as a paradigm or a way of thinking. Like the respondents say this could lead to terrible architectures and spaghetti structures could appear again, the thing we want to avoid. But the respondents also say that often are these vendors sorted out with the help of standards. We agree with the case study material that this is a reason why to start small.

5.1.3 Responsibility questions

5.1.3.1 Business perspective level

Respondent's answers concerning the responsibility questions differ slightly, one opinion regarding service responsibility is that it is up to the organization to decide, it is not up to IT. The IT divisions' responsibility is to provide support for information flow and function, but the absolute responsibility is up to the organization and management.

The responsibility is an organizational question. There are always different roles within a company that owns a business process, because a business process does not exist by itself. The person who executes the service should be responsible, in other word, the person who has deployed the business process or the person who owns it. This might require rethinking in the organization. Furthermore, there are the commercial side, someone sell a service which is a part of a larger enterprise process, and of course everyone wants to own their own service. Responsibilities are related to line of business, organizational and jurisprudence.

A different opinion is that responsibility differs according to organizational level, at business level there are services that can be derived to a specific process, e.g. a credit control implemented as a SOA service. Because of the business perspective there must be some one responsible for the service at this level. What is the service designed to do from a business perspective? There are also the underlying architectures, as the middleware layer; here someone must be responsible for the service depending on the organizational structure. In short terms; there has to be responsibilities and ownership at each level in the organization, but especially at the business level, otherwise one simply move it to the IT division and then there might be problems, a similar problem that we have today with architectural spaghetti structures.

One important issue concerning handling services is to define who is responsible for the service during its lifetime, when this is decided we believe that many problems can be solved, maybe a service registry can deliver those demands. But also making it easier to handling services and truly knowing who is responsible. The National Office of Pollution Prevention et al (1997) state that government; business and non-governmental organizations can apply the life-cycle concept to their decision-making processes related to environment and product policy, design, and improvement. The life-cycle approach can also be used as a scientific tool for gathering quantitative data to inventory, weigh and rank the environmental burdens of products, processes and services.

All organizations make decisions that can influence the inputs and outputs of upstream and downstream stages of the life cycle. We think that one of the essential pieces for questions concerning responsibility and managing identified services is to recognize what information you need for supplying a specific service, the requirements of the service, what input and output should the service provide and simultaneously someone who delivers this. This creates a responsibility matrix where it is visible which part is owned by whom. We also agree with the material based on the case study that controlling the intellectual assets is important so one does not develop solutions that already exist. Once again the importance with service life cycle management, the business decision-making approach which considers benefits, costs and risks over the full life cycle of a service and considers the product life cycle as a whole and optimizes the interaction of product design, manufacturing and life cycle activities.

The respondents refers to service reference library to managing services because in the future there will be many different services; business services, technical services, services in many different shapes. It is important in relation to business services to know which roles there are, what the organizations requirements are, if there are external participants, if there are agreements involved and if there are changes in the market to expect e.g. governmental rules and regulations such as the law of personal records. This is in line with what we studied in the theoretical frame of reference; a service registry. Owens (2006) state that service registry is necessary in a SOA to avoid these problems. Ort (2005) says that service registry is not only a registry solution; it's also a cornerstone of SOA governance. It gives organizations a registry for publishing and discovering needed information about services. According to Fox (2006) it gives organizations a centralized way of tracking and managing information about services and related content based on things like life-cycle stage and organization policies. Furthermore, we would like to point out that it is necessary that organizations operators sit down and inspect the enterprise processes with IT architects and investigate what information is essential.

We agree with the respondents when they say that business side uses technological definitions of a service, but instead they should be concerned with what the service provides and what to put in to the service to get the best result. The importance of the governance questions is to push for ownership around master data. For example implementing large systems by the book involves a lot of effort about securing master data and defining it the right way. If this is done properly before realizing the solution the processes and applications other solutions will function.

It is important to control the right things, the respondents believe that the level of governance should be high concerning IT, but governance can only be high levelled if one has succeed and made it right, it will be more difficult if the development is wrong. In addition to that, it is important when steering projects that line of business is represented as well as provided with requirements from the beginning.

5.1.3.2 Architectural perspective

We think that the responsibility questions are the most difficult to resolve, also we believe that this is the question that can assist avoiding architectural spaghetti. When organizations truly figures out who is responsible for a service, its lifetime and were it might be, the problems can be solved. The theoretical and the case study material show that the responsibility for a service is a question of agreement. E.g. if a company for example buys a service from a service provider, the service provider is therefore responsible for the service. The service provider is responsible for the service and the application, the infrastructure, the power and network within the service. Almost certainly is it the service provider who is responsible for all delivered components.

IBM as an outsourcing supplier or provider might be responsible for the deliverance of a specific service regardless being in possession of the application or the hardware in use. But they could in fact be responsible for the quality of the service. It is actually simple, line of business buys a service from a service provider, and the service supplier therefore is responsible for the delivery of the service. But this is not obvious, traditionally speaking, in the role as an outsourcing company, they are responsible towards customer for the IT delivery and one can have several different actors. If we do not know who is responsible for what, the result can take us back to the problem with spaghetti structure. If one spread services between

different suppliers, then the organization must have an architect who handles and communicates regarding the structure to avoid spaghetti. This can be much disorganized.

The respondents' opinions are, like stated before, that the responsibility for a service is actually shared, a service has different owners or the different parts of a service have different owners, e.g. the subscriber of a service or the buyer of a service and this authority has responsibility for the service existence. When it comes to maintenance and extensions of the service etc. it is a shared responsibility between actors and divisions.

The information needed for a service is different depending on who responds to the question and which his/her professional role in the company is. The information an infrastructure supplier might need, is comprehension of desired transaction quantity e.g. different orders, invoices etc. Information about accessibility and response time are also important. For example, is there call for service maintenance or should the service be active twenty for seven? This bears a resemblance to the supplier of the application who needs to know what and how to design the service. Respondents say that there is actually no difference from how one does today. One needs to know the impact on the business if the system should crash and prioritize in problem situations. Geographical information of the users is also essential.

Another important opinion is that the information about each service will vary depending on the stage, the design stage, implementation stage and operation stage. But most important, the classic fundamental part in architectural contexts is the objective, why shall we have this service, what function does it provide and what is our need? To satisfy the organizational need one must know what the actual need is, what is the objective and is it worth it? When the organization knows this, the rest will pursue.

How to make a service visible is associated to the organizational structure and culture, how the organization works and how organizational questions are handled. It is important to study the enterprise design. The world of enterprises has focused on business units instead of market segments. Presume you have a number of organization areas or a number of different companies or sectors within an enterprise. To manage services in the enterprise it is important to concentrate on the requirements, not on information technology or the end in itself, instead focus on functions. Can a service registry reply to all the demands about a knowledge regarding service, or do we need more to handle service on different levels. This is not just about technology, it is mostly about knowing the organizations different services, once again we want to point put governance and management around services at every level; business, architectural and technical.

5.1.3.3 Technical perspective

The technical answers to questions with reference to responsibility incline towards shared responsibility like the other perspectives. During the development time it is always the development project that is responsible for the service. When they are finished it will be moved to an administration unit, when the service is put into operation the administration unit is responsible.

We think that it is significant that services, applications and composed applications which use services, have a life cycle. The respondents say that it has shown to be easy writing an application but it can be torture to maintain it, moreover to delete it, just because no one knows what the service is developed to execute or when it has fulfilled its purpose. Life cycle management is very important in relation to handling services; approximately they have to be

treated as applications, much like the product life cycle, we agree with this and refer to several early statements. As a marketing planning tool, PLC helps to outline potential marketing strategies. However, applying PLC concepts to predicting is not particularly useful. Critics claim that sales histories are too diverse, and that stages vary in length for different products. Thus, while it has its purpose, PLC concepts ought to be applied with care (Rea & Kerzner, 1997).

The case study shows that one must not forget the master data, the foundation of a service; the data is often the forgotten child in SOA. If one has a high level of abstraction concerning a service, there are always several data sources which are used to create high level. It is necessary to know if one can trust the data in the source. For example if there is a customer registry which was based upon three earlier customers registers as a result of buying several companies. How does one know if this customer register is better then the earlier, e.g. if there are three customers called John Smith, is it possible that this is the same person or is it three different persons? Maybe it is the same person, but with different addresses? If one mixes, in this case customer registers with something else, one get nonsense. Much like mixing ice cream with fertilizer, what you get is fertilizer.

We can establish that there are several questions to be answered when figuring out what one needs to know about a service. Which service do we need? What are the demands on the service provider? What is the cost, answering time, availability, security etc? What service level agreements does the service contain? It is of most importance that customers dare to use the service, we also need to know that it is the right service, an adequate security level and that the service upholds the expectations, demands and functionality it provides. Through negotiation between the IT service unit and business units, a service level agreement leads to articulation of the services IT offers and the costs of the services. These dialogues clarify the requirements of the business units, thereby informing governance decisions on infrastructure, architecture, and business application needs.

5.1.4 Informational requirement questions

5.1.4.1 Business perspective level

Informational questions about dissemination information regarding services will be improved; stated the respondents, because no one likes to work in vain. One way to spread information about service existence within an organization is to apply a service catalogue at a portal in the organization, were one handles the services. Several times in our case study the word catalogue has occurred, we opted to call this registry, but this goes by numerous names e.g. repository, registry etc. We, as well as the respondents agree with Windley (2006) and Abrahams (2005) when they stated the aim of using a service registry, it is suitable to investigate who controls SOA metadata, and ask questions such as; who is responsible for the service and the service life cycle? Who are developers, the IT operations people, the Line of business people and which are the SOA committees?

We believe that controlling and handling services in SOA partly involve asking a lot of questions, we support this assumption on earlier statements by respondents and the theoretical frame of reference. Everything is of course dependent on the type of organization, structure, culture etc, and which competence level the organization has. Most important is that the information about services is accessible and visible, absolutely not hidden.

5.1.4.2 Architectural perspective

This perspective concerning informational questions about spreading information regarding services is that one should have some sort of decision forum were both business and IT architects are represented. The respondents do not believe that one can spread information through an intranet; spreading information is an organizational decision to make. It is also essential to have hard governance from the management, it is the only way. The decision forum could resemble a registry which is our own thoughts, but this assumption can be based on earlier statement but mostly on our on ideas of the importance of spreading information regarding services and making services visible to the organization.

The respondents also believe that by trying to change the behavior and culture of current processes in an organization and shifting the mindset of the individuals, by introducing new ways of acting and thinking efficiently, would basically increase a 'natural' need for new services. We think that culture and behavior is a cornerstone in business processes like the respondents says. It is important when implementing new architectures, the changes it brings will involve the organizational culture and this is the time to improve current processes.

5.1.4.3 Technical perspective

The respondents say that spreading information about services today works fine with e-mail, but this might lead to spaghetti and difficulties with version handling. Thus it is not specifically used for services but the same thing concerns how one spread information about a system or a specific project. If one look at huge organizations or e.g. Internet, the respondents believe that service registry is a good way to begin, a flexible registry so that one can search for the services. There must be good search engines to find services. There are UDDI;s that acts like public registries. For example when one of the respondents told us that he needed a service to send SMS from a website to cell phones. He found many different services that corresponded to his demands but no one was functional. Public UDDI;s does not seem to have entirety functionality and quality. Service registry is one way to spread information, much like the organization who handles information concerning telephone numbers and public records; they market their services to be visible. In the same way one shall market the service registry to internal projects. Owens (2006) state that one thing SOA needs is a registry.

In addition to service registry, exactly which kind or at which level it should be is not easy for the respondents to answer. Concerning a quantity of service registries it is of most importance that services can find each other. It is not necessary of the reason that someone need to use a service or at high level develop a service. Furthermore to comprehend what the services actually do or which processes the services belong to or what the reason of the service were in the beginning.

We assume that the essence of SOA is to control and manage services, as we stated many times before in this analysis, how to technically deliver such a registry we can not describe, but we have in this discussion gained a clear picture of what a service registry should be based on and what a SOA strategy overall should be founded on.

5.1.5 Governance questions

5.1.5.1 Business perspective level

Business perspective delivers four main components in a governance strategy for SOA; one is governance around the applied technical architecture, do we use the right products, decisions around e.g. technical platform etc. Technically there will be several alternatives to do this, but

does the organization know what they want to accomplish, have everyone given their commitment, then the identified problems will be solved in every project shape. Second, governance concerning master data is of importance, there has to be strong governance around how one handles master data.

Governance also controls the investment costs, this is a way of securing the principles one want to use. It is difficult but it is important to stretch one's "tentacles" far into the organization. It is also a matter of education, securing the correct competence and that the employees have the right knowledge of how the organization wants to work. Education and securing competence lives at governance level. Governance is a framework and contains a lot, if an organization can succeed in their governance work then everything will pursue fine. These four components; technical architecture, master data, finances and education should be steered inside a governance function.

The theoretical study and the case study show that SOA governance is an important part, management and control, it is also a strategy were to head and why, the foundation might e.g. be that one want a rapid organization that can meet changes on the market. Another part that's important is security.
The motive when one first started to talk about Web services and why it has not become so successful is the thought that one could just sell Web services as a product out of a commercial perspective.

In the case study at business perspective level also shows that a SOA strategy also includes an implementation plan; you don't make a SOA implementation in a day. It will take much time; in the strategy one has goals and objectives, and to achieve the goal one has tactics, tactical decisions to reach the goals. These can change over time depending on context e.g. changes on the market, buying of companies, selling of company divisions that influences tactic and strategy.

We believe that it is of great significance that several building blocks are involved in a governance strategy for SOA, fundamental building block as IT management, life cycle management, service management, IT governance, SOA governance etc. But most of all it is essential to bridging the gap between line of business and IT. IT management deals with questions relating to organization of IT, responsibilities and authority, furthermore to handle or avoid risks that evolve and might endanger successful management. Just as Magoulas and Pessi (1998) state that IT management includes handling problems, it is important to use IT management when bridging this gap, to speak the same language, this has been preached for several years, perhaps now it is fundamental to really use well-planned strategies and management to reach success with this new paradigm, learn from earlier mistakes this time, do not invent the wheel once again.

5.1.5.2 Architectural perspective

One of the building blocks that are involved in an adequate governance strategy for SOA is clear business requirements, how to get this is difficult to say state the respondents. But the organization always has business strategy and business architecture. If the organization has a clear definition why they are on the market this will lead to better decisions around the IT architecture. This is for the most difficult for the IT division to accept; why the organization exists.

On the basis of the business architecture one need to define what support is needed to be competitive, a clear business plan is essential and we assume that every organization today have a business plan, more or less clear. The problem is that the business division does not communicate with the IT division in terms of business demands. It is a huge responsibility for the IT architect to translate the business demands into a comprehensive language. As well as it is for a business architect who put together the processes to define business processes and than convert the processes into IT terms, close collaboration between the levels and communication is essential.

Automatic and dynamic are different things depending on definition. Governance is one part of this, and of course must there be rules and regulations. Otherwise one might move the complexity up to another level and closer to the user and we have a new problem, similar to the old ones. Consider a room, in one corner we have bare iron, (Hardware, discs, "the brain" etc), and in one corner we have data storage, metadata, in the third corner we have network, communication and integration and in the last corner we have the functions (the services needed). How do I connect these four corners so that I can connect anything based on which demand I have? The fundamental parts according SOA are; infrastructure and some kind of need and purpose, some realization layer and an enterprise integration bus. There are a lot of important parameters in SOA but we believe that one of the most important is communication. Without communication there will be no information exchange and the system will be worthless. But how do we ensure that we have the right approach concerning connection between systems and subsystems within the context of SOA? Systems should be connected to promote a high degree of independence or as we would like to call it loose coupling. When building architecture based upon loose coupling, is it of importance to have a well designed, independent communication system that only transfer data and information between systems and do not care about the changes of information in subsystems. Communication within SOA is based upon accepted open standards and if properly designed, the communication system will be independent, valuable and not affected when the business changes their focus or redesigning their strategy.

5.1.5.3 Technical perspective

The thoughts about governance are slightly different regarding to the technical perspective. The conviction about importance of governance does not exist fully. For example look at Internet, how did Google start; two guys in a garage and how did Bill Gates begin; in a garage. But take a look at UDDI, it begun with control, management by rules and regulations how to develop and how to use. But this is an example that it does not work and no one uses it. Google, on the other hand is used by many.

It is not always relevant to have management and control from above in a company, however, companies want to have control over their assets, but in many cases this can impede development. Take a look at Google; it is a company that has been very successful in modern time. They work in a little bit different manner, four days a week they work in collective projects but the fifth day people can arrange their own projects. This is maybe not a well-known fact, but most of Google's employees have their own hobby projects parallel to their work, this is how Google Earth was developed and became product.

Other cornerstones for a governance strategy for SOA are knowledge and comprehension, learning by doing in small projects and it is very important that the business division is involved. It is of importance that one work with open standards even if the company should choose to implement SOA in locked architectures from a vendor e.g. one thing one want to

address is the ability to fusion companies and then there is an opportunity that one doesn't use the same vendor and one get trouble.

Many people talk about the importance of SOA governance, that it is important to include SOA governance in the organizational strategy, but what SOA governance should contain is more difficult to determine. We think that from one point of view SOA governance provides context for system architects and designers, it provides a framework to ensure that IS/IT, and in this case SOA is aligned with the business, we believe that SOA governance can help organization s to easier facilitate business and maximize benefits. The principles of SOA governance makes sure that assets are used in a more responsible way and risks are managed more suitably. According to us organization s have much to win by adopting SOA governance; by managing the infrastructure properly organization s will get better alignment with attention on business and more understanding among employees concerning Service Oriented Architecture.

If we see SOA governance from a slightly more "soft" viewpoint SOA governance is about achieving a desirable level of simplicity concerning decision rights and responsibility spheres (Ross & Weill, 2004). SOA governance can help organization s to clear out ownership and responsibilities and increase shared understanding among all stakeholders based on a common language. One must not forget that SOA governance is a tool and an extension of IT governance, the principles of IT governance are adjusted to the context of SOA and an organization can not only apply a SOA governance strategy without an overall IT governance plan.

The use of IT and governance can help these enterprises in their activity and managing information. Building your decision process based on governance theories, minimizes the risk to make bad (misleading) decisions. High-quality IT governance harmonizes decisions with desired behaviors and business objectives. Without carefully designed and implemented governance structures, enterprises leave this harmony to chance.

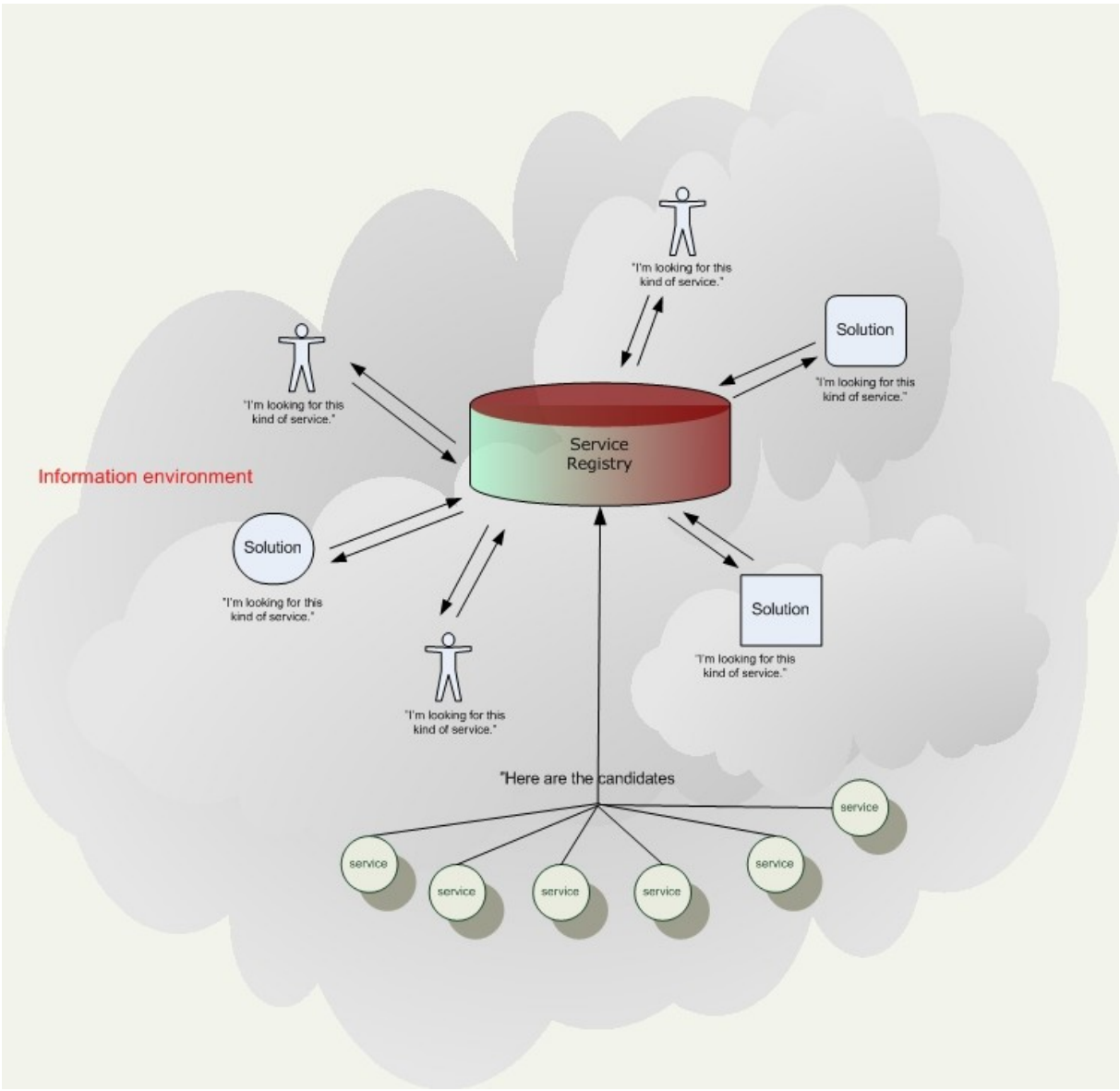


Figure 14. Our vision of a Service Registry

6 CONCLUSION AND RECOMMENDATIONS

Our concern was when using SOA without consideration about earlier problems with architectural spaghetti a similar problem might lead to another level of spaghetti complexity, we choose to call it; service spaghetti. Which key success factors can be identified for managing balance of dependencies in changing information environments characterized by SOA? We had moreover a mission to increase the knowledge of SOA; we believe that with this master thesis we contribute to improved knowledge of SOA and that we truly raise facts regarding details in a successful strategy for SOA. Our conclusions and recommendations are all based on our theoretical frame of reference, the result of the case study, and our own thoughts in the analysis and discussion.

We came to the conclusion that it is important for organizations to have a service registry. We do not concern how many registries there should be within an enterprise just the fact that it is important for organizations to have a registry for services. It is essential that the organization from the beginning build in a service registry into the architecture. A service registry must not become something like a management information system (MIS); were every service is stored in a central registry. MIS;s have shown too be inadequate and instead of one central service registry there might be decentralized registries within an organization and divisions must be responsible for their own registry and control and manage their own services. We call it Decentralized Service Registry Management (DSRM) and services registries will be able to interact with each other and exchange registry information based on e.g. demand and responsibilities.

The world of business and organizations must understand that SOA is not a product it is a way of thinking and doing, it is more a method than nothing else and organizations must ignore the fact that one does not buy SOA as this new revolutionary product that will fix all their integration problems. We have also concluded that organizations must realize that a SOA implementation will take time and they must not rush into something just because of the hype.

Services must evolve from business perspective and not from IT and this will positively bring line of business closer to IT and they will be able to talk to each other in a more natural comprehensive way and bring the cap between each other. We recommend that organizations interested in SOA first develop strategies, principles and policies for both IT and SOA governance. This will help them to identify services at business level and not IT. Organizations must think in strategic architectural terms and first distinguish the business and business objectives. It is essential to create control mechanisms for behaviors and culture concerning projects and designate people to influence concerned actors to adopt SOA, service reusability and service registry as a fundamental part of design. Furthermore make sure to develop a positive personal attitude among employees; everybody must be committed, from business people to the developers, from stakeholders to architects, identification of expected requirements and the employees need to learn the new methods and terminology for success.

Technical developers need to look of today and see what can be reused, instead of writing new applications; this is a significant part to avoid spaghetti structures. It is essential to recognize which information is needed for supplying and managing a specific service, elucidate responsibility areas and we also recommend control of the intellectual assets so one

does not develop solutions that already exist. With this it is also important to agree upon standards because everyone have to use the same language and tools for communication.

Implementing a Service Registry which gives organizations a central or regional way of tracking and managing information about services and related content based on things like service life cycle stages and organizational policies. We recommend that organizations render a common language for communication between business side and IT and speak of service definitions in an understandable way. Instead of using a lot of technical terms that only the technicians understand, use standards; it will make SOA successful. Furthermore organizations have to understand the importance of SOA governance strategies for control and management concerning defining and securing master data in the right way. If this is done properly before realizing the solution the processes and applications among other solutions will function.

6.1 Key success factors

We complete this master thesis by listing key success factors concerning managing balance of dependencies in changing information environments characterized by service oriented architecture, the listed key success factors are all founded on our theoretical frame of reference, the result of our case study at IBM, and our own thoughts;

- **SOA is a way of thinking and doing - it is a method not a product**
- **Build in a Service Registry into the architecture from the start**
- **Services must evolve from business perspective and not from IT**
- **Create a positive personal attitude among employees**
- **Apply Service life cycle management - clarify service ownership**
- **Apply SOA governance**
- **Enclose Master data definition for Quality dimensions**
- **Promote improved communications**
- **Bridging the gap between business an IT**
- **Agree upon standards**

7 REFERENCES

Abraham, Peter (2006). UDDI inadequate for SOA. *IT-Director.com* 23 Sept. Available: <http://www.it-director.com/article.php?articleid=12909> (2006-03-01)

Abrams, Charles, Valdes, Ray & Mitchell Smith, David (2005). Core Web service standard UDDI evolves with version 3.0.2, *Gartner group*, 8 February.

Ackoff, Russel, L. (1967). Management Misinformation Systems. *Management Science*. Volume: 14, No: 4 Pages: 147-56, December.

Albrecht, K. (1988). *At America's Service.* Dow-Jones Irwin, Homewood, IL.

Andrews, Whit. (2004). Be vigilant when adopting Web services standards. *Gartner*, 23 Nov.

Ansoff, Declerck & Hayers (1976). *From Strategic Planning to Strategic Management*. John Wiley & Sons.

Arkin, A, Askary, S, Fordin, S, Jekeli, W, Kawaguchi, K, Orchard, D, Pogliani, S, Riemer, K, Struble, S, Takacsinagy, P, Trickvic, I & Zimek, S. (2002). *Web Service Choreography Interface 1.0*, August. Available: <http://www.w3.org/TR/wsci> (2006-04-03) Available:http://www.devshed.com/c/a/Web-Services/Introduction-to-Service-Oriented-Architecture-SOA/2/ (2006-05-26)

Arsanjani, Ali, Ph.D, (09 Nov 2004) *Service-oriented modelling and architecture, How to identify, specify, and realize services for your SOA.* (Elektronisk) Available:<http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/index.html> (2006-06-12)

Bainchini, Devis, De Antonellis, Valeria, Pernici, Barbara & Plebani, Pierluigi (2005). *Ontology-based methodology for e-service discovery.* Science Direct Information Systems 31 (2006) Pages: 361-380. Accepted 20 February 2005.

Baker et al, (2003). *Resource Discovery Network*, Publications. 9 October. Available: <www.rdn.ac.uk/publications/rdn-ltsn/types/> (2006-05-26)

Barlow, E.R, Putnam, A., et al. (1966). *Samordnad företagskontroll – totala informationssystem I praktisk belysning*. J Beckmans.

Barone, C.A. (1989). Planning and the Changing Role of the CIO in Higher Education, *Information Management Review* Volume: 5, Pages: 23-31.

Beck, U. (1992). *Risk-Society*. London: Sage.

Benney, M, & Hughes, E. C. (1970). *Of Sociology and the Interview, In Sociological Methods: A Sourcebook*, edited by N.K. Denzin, Pages: 176-181, Chicago: Aldine.

Benney, M. & Hughes E.C. (1970). *Of Sociology and the Interview. In Sociological Methods*: A Sourcebook, edited by N.K. Denzin, 175-181. Chicago: Aldine.

Bianchini, Devis, De Antonellis, Valeria, Pernici, Barbra & Plebani, Pierluigi (2005). Ontology-based methodology for e-service discovery. *Information Systems*. Volume: 31, Pages: 361-380. 20 February. Elsevier B.V.

Box, J. (1983). *Extending product lifetime*: Prospects and opportunities, European Journal of Marketing, Volume: 17, Pages: 34-49.

Brancheau, James C. (1997). *The evolution of Information Systems Management*: *Speed, Flexibility, and Responsiveness for the New Millennium*, University of Colorado, August.

Carlson, Brent & Marks, Eric (2006). *SOA Governance Best Practices – Architectural, Organizational, and SDLC Implications; Taking the management of services to the next level.* SOA Web Services Journal. Available: <http://webservices.sys-con.com/read/175376.htm> (2006-05-26).

Castells, M. (2000). *The Rise of the Network Society* (2nd edn). Oxford: Blackwell.

Central storage of policy and metadata, SOA software (2006). Available*:* <http://www.soa.com> (2006-03-29).

Channabasavaiah Kishore, Holley Kerrie & Tuggle Jr, Edward. (2003) *Migrating to a service-oriented architecture, Part 2, Introduction and overview continued*

Available;<http://www-128.ibm.com/developerworks/webservices/library/ws-migratesoa2/index.html> (2006-06-09)

Chatarji, Jagadish (2004). *Introduction to Service Oriented Architecture (SOA).* Web Services. Available: http://www.devshed.com/c/a/Web-Services/Introduction-to-Service-Oriented-Architecture-SOA/ (2006-05-26)

Checkland, Peter (1981). *Systems thinking, systems practice*. Great Britain: John Wiley & Sons Ltd.

Christensen, E. Curbera, F. Meredith, G & Weerawarana, S. (2001). *Web Services Description Language (WSDL) 1.1, World Wide Web Consortium.* W3C Working Group Note. Available: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315> March 2001. (2006-03-17).

Ciccarese, Paolo, Caffi, Ezio, Quaglini, Silvana & Stefanelli, Mario (2005). *Architectures and tools for innovative Health Information Systems*: The Guide Project. International Journal of Medical Informatics, August. Volume: 74, Issues: 7-8, Pages: 553-562.

Cordeiro, Maria Inês & de Carvalho, Joaquim (2002). *Web Services*: *What they are and their importance for libraries*. Emerald Insight, December, Volume: 32, Issue: 4, Pages: 46-62.

Cwiklo, Bill (2000). As The Internet Matures; Litigation Support Options Become More Sophisticated. *Law Technology News*. Available: <http://ltn-archive.hotresponse.com/february00/case_management_p24.html> (2006-02-09).

Day, G. (1981). *The product life cycle*: Analysis and applications issues, Journal of Marketing, Volume: 45, Pages: 60-67.

Dijkstra Edsger W. (1974). *On the role of scientific thought*. 30th August, The Netherlands. Appering in EW Dijkstra, Selected Writings on Computing: A personal perspective, Springer Verlag, 1982. Burroughs Research Fellow.

Drucker, Peter (1974). *Management*: *Tasks, Responsibilities, Practices*. New York: Harper & Row.

Duberley, Joanne & Johnson, Phil (2003). Understanding management research: *an introduction to epistemology.* London: Sage Publications Ltd.

Dubray, Jean-Jacques (2005). *Fundamentals of Service Orientation*. Attachmate Corporation February. Available: <http://www.attachmate.com/NR/rdonlyres/9AB80055-9428-4453-9F7C-A1A2A7C32FAE/0/tp_ssb_SOA.pdf> (2006-04-20).

Earl, M.J, (1989). *Management Strategies for Information Technology*, Prentice-Hall.

Easterby-Smith, Mark, Thorpe, Richard & Lowe, Andy (2002). *Management Research An introduction 2nd edition*. London: Sage Publications Ltd.

Effective SOA deployment using an SOA Registry Repository, A practical Guide, September 2005. Available: <http://www.sun.com/products/soa/registry/soa_registry_wp.pdf> (2006-03-29).

Erl, Tomas (2005). *Service Oriented Architecture, Concepts, Technology and Design*. Prentice Hall Professional Technical Reference.

Evans, Philip B & Wurster, Tomas (2000). *Blown to Bits; How The New Economics of Information Transforms Strategy,* Harvard Business School Press, Boston, MA pp.31

Fayol, H. (1916/50). Administration Industrielle et Generale. Paris: Dunod.

Flick, Uwe (2002). *An Introduction to Qualitative Research.* Second Edition, Sage Publications Ltd.

Fox, Steve (2006) Dilbert kan SOA. *Computer Sweden*. 22 March.

Galliers, Robert D & Leidner, Dorothy E. (2003). *Strategic Information Management: Challenges and Strategies in Managing Information Systems*. Third edition. Great Britain: Biddles Ltd.

Gavan, D. (1984). *What does product quality mean?* Sloan Management Review.

Geertz, C. (1983). *Local Knowledge: Further Essays in Interpretative Anthropology*. New York: Basic Books.

Glazier, Jack D. & Powell, Ronald R. (1992). *Qualitative Research In Information Management*, Libraries Unlimited, INC.

Grembergen, Van, W. Introduction to the Minitrack: IT Governance and its Mechanisms. 35th HICSS conference. Available: <*http://computer.org/proceedings/hicss/1874/tracks8/187480242.pdf*> (2006-03-28).

Grolmolski, B, Grigg, J & Potter, K. (2001). IT *Spending and Staffing Survey Results.* Gartner R-14-41158, 19 September.

Gronross, C. (1986). *From scientific management to service management: a management perspective for the age of service competition*. International Journal of Service Industry Management Volume: 5, Issue: 1, Pages: 5-20.

Habermas, J. (1996). *Der Habermas Reader.* Cambridge: Polity Press.

Hammer, Michael & Champy, James (1993) *Reengineering the Corporation, A manifesto for Business Revolution,* Harper Business.

Handy, C. (1989). *The age of Unreason*. London: Business Books

Handy, C. (1996). *Beyond Certainty: The changing Worlds of Organizations*. London: Arrow Books

Harding, Chris. Dr. (2005). *The Key to Success with SOA*. 31 July. The Open Group. Available: http://www.ebizq.net/topics/soa/features/6151.html (2006-05-26).

Hewitt, Carl & de Jong, Peter (1982). *Open Systems,* Artificial intelligence laboratory, Massachusetts institute of technology.

Hewitt, Carl & de Jong, Peter (1983). *Analyzing the Roles of Descriptions and Actions in Open Systems,* Massachusetts Institute of technology, artificial intelligence laboratory.

Hodge, Andrew (2005). *Harnessing an SOA to assist with Continuous Business Process Improvement*. Australian Bureau of Statistics.

Holme, Idar Magne and Krohn Solvang, Bernt (1997) Forskningsmetodik. Om kvalitativa och kvantitativa metoder. Lund: Student literature

Hradil, S. (1992). Zwischen Bewußtsein und Sein, Opladen: Leske and Budrich.

Hugoson, M.-Å. (1986). *Informationssystem i ett verksamhetsperspektiv. Funktionell strukturering*. Nordisk DATAnytt, Nr 10.

Höij, Magnus (2006). Inget processarbete utan bra grund. *Computer Sweden*. 22 March.

ICHNET (2005). *Interoperability Clearinghouse Glossary of Terms.* Available: <http://www.ichnet.org/glossary.htm.> (2006-05-25)

IT Governance Institute (2003). *Board Briefing on IT Governance*, 2nd edition. Rolling Meadows, USA.Available:<http://www.itgi.org/Template_ITGI.cfm?Section=ITGI&CONTENTID=6658&TEMPLATE=/ContentManagement/ContentDisplay.cfm> (2006-04-20)

Java Sun (2006). Available: <http://www.sun.com/products/soa/index.jsp> (2006-03-14)

Jazayeri, Mehdi, Loss, Rudiger G.K. & Musser David R., (2000) *Generic Programming*. International Seminar on Generic Programming Dagstuhl Castle, Germany, April/May 1998, Selected Papers. (Lecture Notes In Computer Science; 1766) Springer Verlag, Berlin Heidelberg.

Johnston, Gary P. & Bowen, David V. (2005). *The benefits of electronic records management systems: a general review of published and some unpublished cases*. Records Management Journal. Volume: 15 Issue: 3 Pages: 131-140.

Kakabadse, Nada Korac & Kakabadse, Andrew (2001). I*S/IT governance: Need for an integrated model, Corporate Governance* Volume: 1,4, Pages: 9-11, MCB University Press 1472-0701, Emerald Insight.

Khosrowpour, Mehdi (1997). *Managing Information Technology Resources and Applications in the World Economy*. Idea Group Inc.

Kingstedt, Anders (2005). *SOA from Data to Process – concerning bridges and others*. KommIT 17 Nov 2005. Available:<http://www.knowit.se/KIT_templates/Page____1202.aspx> (2006-05-26)

Kotler, Philip & Joanne Scheff (1997). *Standing Room Only - Strategies for Marketing the Performing Arts.* USA: Harvard Business School Press.

Kotler, Philip (2000). *Marketing Management*. 10th ed. Upper Saddle River NJ: Prentice Hall.

Kruchten, P. (2002). *The rational unified process: en introduktion*, swedish edition. London: Addison-Wesley.

Lawrence, P. (1986). *Invitation to Management*. Oxford: Blackwell.

Levitt, T. (1965). *Exploit the product life cycle*, Harvard Business Review. Volume: 43, November-December, Pages: 81-94.

Lübcke, Poul et al. (1988) Filosofilexikonet - Filosofer och filosofiska begrepp från A till Ö. (red) J. Hartman. Stockhom.

Magnusson, Johan & Olsson, Björn (2005). *Affärssystem.* Göteborg: Studentlitteratur

Magoulas, Thanos & Pessi, Kalevi (1998). *Strategisk IT management*. Diss. Göteborgs Universitet. Göteborg: Studentlitteratur

Magoulas, Thanos, Fil. Dr., Department of Informatics, Göteborg University, Göteborg. Conversation 09:00 – 10:00, May 17, 2006.

Marchetti, C, Pernici, B & Plebani, P. (2004). A quality model for multichannel adaptive information, in: *Proceedings of the 13th International World Wide Web Conference*, Alternative track on Web Services, ACM Press, Pages: 48-54.

Mintzberg, H. (1973). *The Nature of Managerial Work*. London: Harper and Row.

Moingeon, B. & Edmondson, A. (1997). *Organizational Learning and Competitive Advantage*. London: Sage Publications.

Morgan, G. (1996). *Images of Organization*. London: Sage Publications.

National Office of Pollution Prevention & the Hazardous Waste Branch of Environment Canada (1997). *Environmental Life Cycle Management*: *A Guide for Better Business Decisions*. Ecocycle, April 1997. Available: <http://www.ec.gc.ca/ecocycle/en/publications.cfm> Last updated: 2003-02-14 (2006-04-04)

Nickull, Duane (2005). *Answer to your questions,* senior standards strategist at Adobe Systems and chair of the OASIS SOA Technical Committee.

Nonaka, I. & Takeuchi, H. (1995). *The Knowledge-Creating company: How Japanese Companies Create the Dynamics of Innovation*. Oxford: Oxford University Press.

Ohya, M. (2005). *Open Systems and Information Dynamics*. Tokyo, Japan: Tokyo University of Science,

Ort, Ed (2005). *What's New in SOA and Web Services?* October 3, 2005. Available: <http://java.sun.com/developer/technicalArticles/WebServices/soa2/WhatsNewArticle.html #regrep> (2006-02-10)

Ossher, H. & Tarr, P. (1999). *Multi-Dimensional Separation of Concerns using Hyperspaces*. IBM Research Report 21452. April 1999. Available: <http://www.research.ibm.com/hyperspace/> (2006-03-20)

Owen, D.E. (1989). IRM Concepts: Building Blocks for the 1990s, *Information Management Review* Volume: 5, Pages: 19-28.

Owens, Chris (2006). *The importance of a registry for a service-oriented architecture (SOA)* HP Invent, January. Available: <http://devresource.hp.com/drc/technical_white_papers/soa_reg/index.jsp> (2006-02-10)

Parnas D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, Volume: 15, No. 12, Pages: 1053–1058, December.

Peppard, J. (2003). *Managing IT as a Portfolio of Services.* European Management Journal. Volume: 21, No: 4 Pages: 467-483.

Peter G. W. Keen, a leading author and consultant, in the foreword to Devaraj, Sarv & Prahalad, C.K. and Hamel, G (1990). *'The core competence of the corporation*', Harvard Business Review, Volume: 68, Issue: 3, Pages: 79-91.

Proceedings of the Institution of Mechanical Engineers, Part B (2001). Journal of Engineering Manufacture. *Life cycle management and assessment: approaches and visions towards sustainable manufacturing*. Issue: Volume 215, Number 5, Pages: 599 – 626.

Puehl, Stefan (2005). *SOA Governance - Control and Empower Flexible Business and IT in Your Client's Organization*. The AMS Offering Management team, September 29. Volume: 4,

Issue: 1.
Available:<http://w3.ams1.ibm.com/services/ams/global/newsletters/ams_offerings_newsletter_09-29-05.htm> (2006-04-18)

Rea, Peter, J. & Kerzner, Harold (1997). *Strategic Planning*. New York: John Wiley and Sons.

Reiter, Raymond (1982). *Towards a Logical Reconstruction of Relational Database Theory,* In Brodie, M.L., Mylopoulos, J.L., Schidt, J.W., editor, *Perspectives on Conceptual Modeling.* Springer-Verlag.

Robinson, W.S, (1951). The Logical Structure of Analytic Induction, *American Sociological Review* Volume: 16, Pages: 812-818.

Ross, Jeanne W & Weill, P (2004). *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results.* Harvard Business School Press.

Ross, Jeanne W & Weill, Peter (2002). *Six IT Decisions Your IT people Shouldn't Make*. Harvard Business Review, November. Pages: 84-91.

Savarese, Daniel F. (2003). On Optimizing Service Orchestration; Think of creativity to overcome the slower performance of Java-based Web services. Available: <http://www.ftponline.com/javapro/2003_04/magazine/columns/proshop/> (2006-05-26)

Schmeizer, Ronald (2005). *The ROI of SOA.* Document ID, ZAPFLASH-20050127. Available: <http://www.zapthink.com/report.html?id=ZAPFLASH-20050127> (2006-05-26)

Shapiro, Carl & Varian, Hal (1999). *Information Rules: A Strategic Guide to the Networked Economy.* Boston: Harvard Business Scholl Press.

Shaw, C. (1966). *The Jack Roller, 2nd ed.* Chicago: University of Chicago Press.

Sifonis, J.G. & Goldberg, B. (1999). *Corporation on a Tightrope: Balancing Leadership, Governance, and Technology in an Age of Complexity.* Oxford*:* Oxford University Press.

SIG security et al. (1997). *Riktlinjer för god informationssäkerhet SSR97EET,* Studentlitteratur.

Sim, Y W, Wang, C, Gilbert, G. & Wills, B. (2005). *An Overview of Service-Oriented Architecture*. School of Electronics and Computer Science University of Southampton.

Simonsson, Mårten (2005). Full koll på ITG. *IDG*, 22 September. Available: <http://cio.idg.se/ArticlePages/200509/22/20050922142325_CIO530/20050922142325_CIO530.dbp.asp> (2006-03-25)

Skinner, Andrew (1970). *The Wealth of Nations: Books I-III.* New York: Pelican Books.

Sloan, Bernie (1997). *Service Perspectives for the Digital Library; Remote Reference Services*. Collaboration in the Information Society. University of Illinois at Urbana-Champaign. Available: <http://people.lis.uiuc.edu/~b-sloan/e-ref.html> (2006-05-26)

Stephen B. Morris (2006). *Separation of Concerns and BPEL*. Java.net. Available: <http://today.java.net/pub/a/today/2006/02/14/separation-of-concerns-and-bpel.html> (2006-03-20)

Strauss, Anselm & Corbin, Juliet (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques.* Newbury Park, California: Sage.

Sundblad, Sten (2004). Serviceorienterad arkitektur – en översikt. Uppsala: Sundblad & Sundblad.

Taylor, F.W. (1947). *Scientific Management*. London : Harper and Row.

Taylor, Steven, J. & Bogdan, Robert (1998). *Introduction to Qualitative Research Methods: A Guidebook and Resource*, Third edition, John Wiley & Sons, INC.

The Open Systems Database Association (Last modified 2006-04-10) Available: < http://www.osda.org/> (2006-05-30)

Trauth, E.M. (1989). The Evolution of Information Resource Management, *Information and Management*. May. Volume: 16, Pages: 257-268.

Turner, R.H. (1953). The Quest for Universals in Sociological Research, *American Sociological Review* 18:604-611.

Van Bon, Jan (2005). *IT Service Management: An introduction based on Itil*. The Netherlands: Van Haren Publishing.

Van Moorsel, A. (2001). *Metrics for the Internet Age: Quality of Experience and quality of Business*. Tech. rep., HP Labs, also in the Proceedings of the 5th Performability Workshop, September 16, 2001, Erlagen, Germany, July 2001.

Voet, Martin, A. (2005). *The Generic Challenge*. Universal Publishers.

W3C Working Group Note (2004). *Web Service Management: Service Life Cycle.* 11 February. Available: <http://www.w3.org/TR/2004/NOTE-wslc-20040211/> (2006-04-06) (IBM internal document)

Ward, J., Griffiths, P. & Whitmore, P. (1990). Strategic Planning for Information Systems. Chichester: John Wiley & Sons, Wiley Information Systems Series.

WebLayers, Inc. (2005). *SOA Governance, Introduction*. Available: <http://www.weblayers.com/gcn/whitepapers/Introduction_to_SOA_Governance.pdf> (2006-04-21)

webMethods (2006). Benefits of SOA. Available: <http://www.webmethods.com/meta/default/folder/0000008660> (2006-05-26)

Wikipedia (2006). The free encyclopaedia. (Search word: service) Available: <http://sv.wikipedia.org/wiki/Huvudsida> (2006-05-24)

Winder, Richard, E. (1993). *Fulfilling service quality through quality's five dimensions,* Second Annual Service Quality Conference, Atlanta Georgia. Available: <http://www.ldri.com/articles/93fillservqual.html>

Windley, Phillip J. (2006). *SOA governance: The rules of the game.* 19 January. Avaiable: <http://www.infoworld.com/reports/04SRsoagov.html> (2006-04-03)

Woodruff, R. B. (1997). *Customer value; the next source of competitive advantage*. Journal of the Academy of Marketing Science. Pages: 139-153.

Word Net (2006). A lexical database for the English language. Cognitive Science Laboratory, Princeton University. Available: <http://wordnet.princeton.edu/> (2006-05-26).

Yefim, Natis, V. (2003). *Service-Oriented Architecture Scenario,* Gartner Research, Publication, ID Number AV-19-6751.

8 APPENDIX

8.1 Interview questions

General questions (to create overview of the subject)

Question 1 – Service Oriented Architecture
- What is your understanding of SOA?

Why this question?
This question is asked because we what to know what level of knowledge and understanding the respondent has on the subject. What terminology he/she uses and their focus on the subject. Knowledge and understanding is a mirror of concepts.

Question 2 - Service Oriented Architecture
- What fundamental parts make SOA attractive to you? (In your day-to-day work).

Why this question?
This question is asked to examine if the respondent has knowledge in IT governance, SOA governance
Service Management, Services (description, definition, advertisement), QoS, Service Level agreements and the concept Service Registry? On the basis of what they answer, we can create a profile on what they know and their focus on the subject according to their responsibilities.
The respondent's answers help us compare the critical success factors we identified. Our intention is not to do a control check up of the respondent's knowledge, or make them fell inadequate; we want to highlight the concept of SOA and important building blocks. The experienced practicians opinions are important for increased knowledge.

Question 3 - Service Oriented Architecture
-What is your opinion were to start a SOA implementation and why?

Why this question?
We have put out this question to see were different people in a SOA project, people who work with SOA start their work. Do they have the same opinion on were to start, small projects and with key processes!?
Do we get different answers dependent on were in the SOA project they have responsibilities. Conceivable answers can be anything from Top down, bottom up, meet in the middle.

Architectural questions (-spaghetti, motive for SOA registry).

Question 4 – IT management

- **What can we do to avoid the same mistake building architectural spaghetti with Service Oriented Architecture?**
- **Or - How can we handle SOA and not build architectural spaghetti once again? (What is needed in addition to traditional concepts?)**

Why this question?

We want to know if IT people have learned from their mistakes and the history of system development. Control is everything in SOA and without management one could not get control! Is it so that system theories once again have been neglected.

In a SOA because, of the complexity and the loose coupling control and order is everything.

This question is asked because we want to know how the respondent is involved in management/control work. Which parts are in use and find out which parts they think are important for control work.

The result from the interviews will work as a road map for comparison between the reality and the theories. This will hopefully increase the awareness around the IT management theories including control.

Responsibility questions (expectations, demands, requirements) (structure)

Question 5 – IT governance

- Who is responsible for a service during its life time?

Why this question?

This question is asked to identify responsibility areas and examine who is responsible for decision making concerning a specific process/service. Which responsibility sphere has each employee, which decisions are they allowed making? SOA is based on partnership, flexibility and security, hence the question.

Were lies the responsibility for specific services.

Information requirement questions (attributes, parameters)

Question 6 – IT management

- What information do you need to know about each service? On what basis do you manage identified service attributes? (What makes it manageable and 'visible'?)

Why this question?

This is the fundamental question for the thesis! How can we increase services visibility, manage services, find services in an enterprise? Information requirement is the basic attribute for all participants, what do they want from the system and how do we supply them with it?

To solve the problem with Service Registry importance.

To see if the respondent, irrespective of work task, gives an answer that is focused on technology and thereby not sufficient.

Question 7 – IT governance
- How can we spread information about service existence to other people in the organization ?

Why this question?
We ask this question to see which people are interested if a specific service exists or not.
Do we have the same problem Java programmers experienced in the beginning, building class libraries? Several programmers developed similar classes around the world, unaware of others developing the same class. We do not want the same problem with services in a SOA, service redundancy.
If we can produce awareness among IT people and increase their knowledge they will gain control of their services. How can we make people more aware of existing information in the enterprise?
Who want a specific service and who can provide the information about that service?
Organizations do not want to spend money on assets that already exist and with assets that are meaningless for the organization.
Business, operative, technical level

Governance questions (limitations, laws, policies, managed information)

Question 8 – SOA governance
- What basic building blocks are involved in an adequate governance strategy for SOA? (SOA governance?)

Why this question?
To see if the respondent is aware of SOA governance as a fundamental part of SOA. This question will bring the interview together.

8.2 Respondent form cover

Interviewer number;	_____
Interviewer location;	_____
First name;	_____
Last name;	_____
Division	_____
Title;	_____
	yy mm dd
Date of interview	☐☐ ☐☐ ☐☐
	hrs min
Time at start of interview;	☐☐ ☐☐
Time at end of interview;	☐☐
Male;	☐☐
Female;	☐☐
	yy
Birth year;	☐☐
Interview in person;	☐☐
Interview by telephone;	☐☐
Interview tool;	☐☐
Tape recorder;	☐☐
Video camera;	☐☐
Computer;	☐☐
Interview carried out in Swedish;	☐☐
Interview carried out in English;	☐☐
Questions in English	☐☐
Answers in English	☐☐
Questions in Swedish	☐☐
Answers in Swedish	☐☐